2023-1765, 2023-1809

United States Court of Appeals for the Federal Circuit

VIRNETX INC.,

Appellant,

v.

CISCO SYSTEMS, INC.,

Cross-Appellant.

Appeals from the United States Patent and Trademark Office, Patent Trial and Appeal Board in No. 95/001,714.

BRIEF FOR APPELLANT VIRNETX INC.

Naveen Modi
Joseph E. Palys
Stephen B. Kinnaird
Igor V. Timofeyev
Daniel Zeilberger
PAUL HASTINGS LLP
2050 M Street, N.W.
Washington, D.C. 20036
(202) 551-1700
naveenmodi@paulhastings.com

Counsel for Appellant VirnetX Inc.

February 5, 2024

SELECTED CLAIMS

U.S. Patent No. 7,490,151

Claim 1:

A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

Claim 5:

The data processing device of claim **1**, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

Claim 7:

A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

Claim 11:

The computer readable medium of claim **7**, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

Claim 13:

A computer readable medium storing a domain name server (DNS) module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) determining whether a DNS request sent by a client corresponds to a secure server;

(ii) when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iii) when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel between the client and the secure server.

CERTIFICATE OF INTEREST

Counsel for Appellant VirnetX Inc. certifies the following:

1. Full name of party represented by me	2. Name of real party in interest represented by me	3. Parent corporations and publicly held companies that own 10% or more of stock in the party
VirnetX Inc.	VirnetX Inc.	VirnetX Inc. is a wholly owned subsidiary of VirnetX Holding Corporation.

4. The names of all law firms and the partners or associates that appeared for the party now represented by me in the agency or are expected to appear in this court (and who have not or will not enter an appearance in this case) are:

None.

5. The title and number of any case known to counsel to be pending in this or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal. *See* Fed. Cir. R. 47.4(a)(5) and 47.5(b).

VirnetX Inc. v. Mangrove Partners Master Fund, Ltd., No. 23-315 (U.S.) (petition for a writ of certiorari docketed Sept. 27, 2023);

VirnetX Inc. v. Apple Inc., No. 22-1997 (Fed. Cir.);

VirnetX Inc. v. Apple Inc., No. 6:13-cv-00211 (E.D. Tex.).

6. Information required by Federal Rule of Appellate Procedure 26.1(b) and (c) that identifies organizational victims in criminal cases and debtors and trustees in bankruptcy cases.

Not Applicable

Dated: February 5, 2024

/s/ Naveen Modi
Naveen Modi

Counsel for Appellant VirnetX Inc.

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Net MoneyIN, Inc. v. VeriSign, Inc.,
545 F.3d 1359 (Fed. Cir. 2008) 46, 49

Panera, LLC v. Dobson,
999 F.3d 1154 (8th Cir. 2021) 43

PAR Pharm., Inc. v. TWI Pharms., Inc.,
773 F.3d 1186 (Fed. Cir. 2014) 49

PersonalWeb Techs., LLC v. Apple, Inc.,
917 F.3d 1376 (Fed. Cir. 2019) 49

Power Integrations, Inc. v. Lee,
797 F.3d 1318 (Fed. Cir. 2015) 32

Stewart v. S. Ry. Co.,
315 U.S. 784 (1942)..39

Transclean Corp. v. Bridgewood Servs., Inc.,
290 F.3d 1364 (Fed. Cir. 2002) 50

U.S. Bancorp Mortg. Co. v. Bonner Mall P'ship,
513 U.S. 18 (1994)..40, 41, 42, 43

United States v. Munsingwear, Inc.,
340 U.S. 36 (1950)..*passim*

United States v. Wilson,
503 U.S. 329 (1992)..35

VirnetX Inc. v. Apple Inc.,
931 F.3d 1363 (Fed. Cir. 2019) ....*passim*

VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.,
778 F. App'x 897 (Fed. Cir. 2019) 32, 45, 54

VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.,
No. 2020-2271, 2023 WL 2708975 (Fed. Cir. Mar. 30, 2023) ....*passim*

VirnetX, Inc. v. Cisco Sys., Inc.,
767 F.3d 1308 (Fed. Cir. 2014) 6, 22, 34, 52

Statutes

Other Authorities

STATEMENT OF RELATED CASES

This is an appeal from an *inter partes* reexamination proceeding before the Patent Trial and Appeal Board ("the Board"). In the proceeding below, the Board issued a decision finding claims 3-4, 9-10, and 15-16 of U.S. Patent No. 7,490,151 ("the '151 patent") patentable but claims 1, 2, 5-8, and 11-14 unpatentable. VirnetX appealed that decision to this Court, and Cisco cross-appealed (No. 23-1809). The Court consolidated those appeals.

There are several other (prior or pending) appeals involving the patent at issue (the '151 patent). In No. 17-1368, the Court considered an appeal from the Board's decision in an *inter partes* review proceeding that found claims 1-2, 6-8, and 12-14 of the '151 patent (and those of a related patent) to be invalid over one of the references at issue in these appeals. This Court vacated the Board's unpatentability findings with respect to the '151 patent as not supported by substantial evidence and remanded the appeal to the Board. *VirnetX Inc. v. The Mangrove Partners Master Fund, Ltd.*, 778 F. App'x 897, 905-08, 911 (Fed. Cir. 2019) (Moore, J., joined by Prost, C.J., and Reyna, J.).

On remand, the Board again found the challenged claims to be unpatentable. This Court affirmed that finding. *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 2020-2271, 2023 WL 2708975, at *9-*11 (Fed. Cir. Mar. 30, 2023) (Stark, J., joined by Moore, C.J., and Hughes, J.). VirnetX filed a petition for a writ of certiorari with the Supreme Court. *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 23-315 (U.S. docketed Sept. 27, 2023).

The ’151 patent was also at issue in an *inter partes* reexamination proceeding No. 95/001,697, initiated by Apple Inc. (“Apple”). In that proceeding, the Board found claims 7, 8, and 11 of the ’151 patent unpatentable. VirnetX appealed that decision to this Court, and that appeal is currently pending as No. 22-1997. The Court designated that appeal and the present appeals as companion cases, to be argued before the same merits panel.

This Court also considered several appeals from district court proceedings that involved the ’151 patent (and related patents). The first appeal (No. 13-1489) was from a judgment by the United States District Court for the Eastern District of Texas in an infringement proceeding initiated by VirnetX in 2010. After a jury trial, the district court entered a judgment upholding the asserted claims against an invalidity challenge, but finding that Appellee Cisco Systems, Inc. (“Cisco”) did not infringe the asserted patents. The jury found, however, that Cisco’s co-defendant—Apple Inc. (“Apple”)—infringed the patents-in-suit, and awarded damages. *VirnetX Inc. v. Apple Inc.*, 925 F. Supp. 2d 816 (E.D. Tex. 2013). This Court affirmed-in-part, reversed-in-part, vacated-in-part, and remanded for further proceedings. *VirnetX, Inc. v. Cisco Sys., Inc.*, 767 F.3d 1308 (Fed. Cir. 2014) (Prost, C.J., joined by Chen, J.).

On remand, after another jury trial, the district court entered a judgment again finding infringement by Apple (Cisco was no longer a party) and awarding damages. In a subsequent appeal (No. 18-1197), this Court affirmed that judgment under Federal Circuit Rule 36. *VirnetX Inc. v. Cisco Sys., Inc.*, 748 F. App'x 332 (Fed. Cir. 2019) (per curiam) (Prost, C.J., Moore and Reyna, JJ.). The Supreme Court denied a petition for a writ of certiorari. *Apple Inc. v. VirnetX Inc.*, 140 S. Ct. 1122 (2020).

This Court also considered an appeal from another infringement proceeding VirnetX initiated against Apple in the United States District Court for the Eastern District of Texas in 2012, asserting the '135 patent (and three other patents). In that proceeding, the district court granted VirnetX a summary judgment on invalidity, finding that Apple was precluded from raising its invalidity challenges because of prior litigation. After a jury trial, the district court entered a judgment for VirnetX finding infringement by Apple and awarding damages. *VirnetX Inc. v. Apple Inc.*, No. 12-cv-00855, 2018 WL 10048706 (E.D. Tex. Aug. 30, 2018). On appeal (in No. 19-1050), this Court upheld the district court's ruling that Apple was precluded from making its invalidity challenges and upheld the finding of infringement as to the '135 patent (and one related patent). The Court, however, reversed the judgment of infringement as to the other two patents at issue, and therefore vacated the damages award and remanded. *VirnetX Inc. v. Apple Inc.*, 792 F. App'x 796 (Fed. Cir. 2019) (Taranto, J., joined by Mayer and Lourie, JJ.).

On remand, the district court held a new trial on damages. The jury awarded VirnetX over $502 million in damages for Apple's infringement of the '135 patent (and a related patent), and the district court entered a final judgment. On appeal, this Court vacated the district court's judgment and remanded with instructions to dismiss the case as moot based on its decision in *Mangrove* (Appeal No. 20-2271). *VirnetX Inc. v. Apple Inc.*, No. 21-1672, 2023 WL 2770074 (Fed. Cir. Mar. 31, 2023) (Stark, J., joined by Moore, C.J., and Hughes, J.). VirnetX filed a petition for a writ of certiorari with the Supreme Court. *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 23-315 (U.S. docketed Sept. 27, 2023).

INTRODUCTION

In the *inter partes* reexamination below, the Patent Trial and Appeal Board ("the Board") found unpatentable several claims of Appellant VirnetX Inc.'s ("VirnetX's") U.S. Patent No. 7,490,151 ("the '151 patent") not on the basis of rejections proposed by the third-party requester in this reexamination—Cross-Appellant Cisco Systems, Inc. ("Cisco")—but on the basis of rejections proposed by a *different* party, Apple Inc. ("Apple") in a *different inter partes* reexamination. Apple, however, was statutorily precluded from challenging these claims in reexamination by this Court's decision affirming a district court's judgment that rejected Apple's invalidity challenge to those claims. Pre-AIA section 317(b) precludes the Board from maintaining an *inter partes* reexamination after "a final decision" that the patent challenger failed to prove invalidity of the challenged claims in a civil case. *See* 35 U.S.C. § 317(b) (2006).

Despite this statutory command, the Board improperly maintained the Apple-initiated reexamination, consolidating it with this reexamination proceeding—an error that this Court corrected in *VirnetX Inc. v. Apple Inc.*, 931 F.3d 1363 (Fed. Cir. 2019). Following this Court's decision in *Apple*, the Patent and Trademark Office ("the PTO") severed the two reexaminations, and terminated the Apple-initiated reexamination with respect to the claims whose validity has been upheld in parallel district court litigation. The Board in *this reexamination*, however, then used the

Apple-proposed rejections—which Cisco never asserted in this proceeding—to invalidate the very claims as to which the PTO has terminated the reexamination in the Apple-initiated proceeding. The Board's decision represents an impermissible end-run around section 317(b) and defies this Court's decisions in *Fairchild (Taiwan) Corp. v. Power Integrations, Inc.*, 854 F.3d 1364 (Fed. Cir. 2017), and *VirnetX Inc. v. Apple Inc.*, 931 F.3d 1363 (Fed. Cir. 2019). This Court should vacate the Board's decision and remand these appeals to the Board for consideration of Cisco's patentability challenges without the improper reliance on Apple-proposed rejections.

Regardless, the Board's decision below addressed claims that have since been held unpatentable and slated for cancellation as a result of this Court's decision in another appeal involving the '151 patent.[1] Because no further action can be taken with respect to a cancelled claim, consideration of the unpatentability of these claims is now moot. Under traditional principles of vacatur, where mootness is caused by a prior court decision preventing review, vacatur is required. *See, e.g.*, *Apple Inc. v. Voip-Pal.com, Inc.*, 976 F.3d 1316, 1321 (Fed. Cir. 2020). The Court should

[1] That decision, *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 2020-2271, 2023 WL 2708975 (Fed. Cir. Mar. 30, 2023), is currently subject to a pending petition for a writ of certiorari filed by VirnetX. *See VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 23-315 (U.S. docketed Sept. 27, 2023). In the event the Supreme Court grants certiorari and reverses or vacates this Court's decision, the patentability of those claims would become live again.

accordingly vacate the Board's decision with respect to claims whose patentability is now moot.

The Board's decision also misinterpreted the asserted references and incorrectly found the challenged claims anticipated or obvious. This Court should reverse or vacate the Board's erroneous rejections with respect to claims whose patentability is not moot.

JURISDICTION

The Board issued its decision on appeal in the *inter partes* reexamination below on June 23, 2020, Appx1-66, denied VirnetX's petition to vacate that decision on September 28, 2020, Appx67-80, and denied rehearing on April 27, 2021, Appx83-104. The Board then issued a decision addressing new grounds of rejection on March 24, 2022, Appx105-117, and denied rehearing on February 10, 2023, Appx118-135. VirnetX timely appealed to this Court on April 7, 2023. Appx2320-2323. This Court has jurisdiction under 35 U.S.C. § 141(b) and 28 U.S.C. § 1295(a)(4)(A).

ISSUES PRESENTED

1. Whether the Board's decision should be vacated because it was based not on rejections advanced in this proceeding, but on rejections advanced in a different *inter partes* reexamination by a party that is statutorily precluded from challenging the claims at issue.

2. Whether the Board's unpatentability findings should be vacated as moot with respect to claims that are slated for cancellation as a result of a decision in another proceeding.

3. Whether the Board's rejections are adequately explained and supported by substantial evidence.

STATEMENT OF THE CASE

I. THE TECHNOLOGY AT ISSUE

The invention at issue relates to secure Internet communications. Computers on the Internet address each other using an Internet Protocol (IP) address, a four-segment string of binary numbers (often represented in decimal form, *e.g.*, "19.28.37.456"). Appx197 (36:61-67). When one device sends data to another over the Internet, the data is broken up into "packets." The packets are labeled with the IP addresses of the sending and destination computers. The packets travel across the Internet using a series of specialized devices called "routers," which direct traffic over the networks comprising the Internet. Appx145 (Fig. 1). Each router checks the packet's destination IP address against internal tables, and passes the packet to the next router. Packets traverse the Internet's hierarchy of networks and routers until they are ultimately delivered to the destination computer.[2]

[2] The Internet's basic operation is succinctly described at Rus Shuler, *How Does the Internet Work* (2002), https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm.

Internet communications are ordinarily nonsecure. Packets can be intercepted during routing by hackers, who can then read the data within. Appx198 (37:11-20). The rapid increase in Internet use in the late 1990s generated demand for easy and secure network communications. Appx180 (1:27-29).

Before VirnetX's inventions, reliable secure communications were primarily achieved through encrypted Virtual Private Networks ("VPNs"). VPNs, however, were cumbersome to use. VPN users had to be trained to set up encryption keys used to scramble and unscramble the data, and the sender and the receiver had to configure multiple parameters for the VPN link. The '151 patent taught an innovative alternative that generated VPNs automatically. The inventors looked to the domain name system, which translates user-friendly "domain names," such as "yahoo.com," into IP addresses that routers can use to direct data to a device, such as "98.137.11.163." For example, when a user types a domain name into a web browser, the user's device sends a request to a Domain Name Server ("DNS"). The DNS responds with the IP address corresponding to that domain name, which allows the user's device to connect with the target computer. Appx197 (36:61-67). VirnetX's inventions allow users to employ familiar techniques—typing a domain name into the browser—whereupon the system seamlessly establishes a secure VPN between two devices. These inventions provided an easy method of ensuring data security, particularly for business travelers or between private networks. *See, e.g.*, Appx181 (3:2-4); Appx182 (6:7-9).

II. THE ’151 PATENT

The ’151 patent relates to a system and method in which a DNS proxy automatically and transparently creates a VPN in response to a DNS lookup. *See VirnetX, Inc. v. Cisco Sys., Inc.*, 767 F.3d 1308, 1315 (Fed. Cir. 2014); Appx197-198 (36:55-38:37); Appx173-174 (Figs. 26-27).

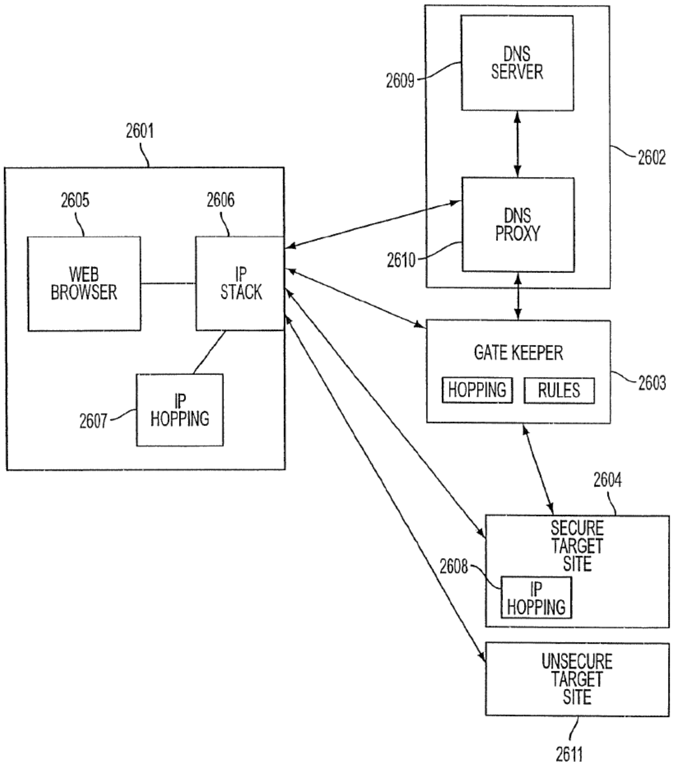


FIG. 26

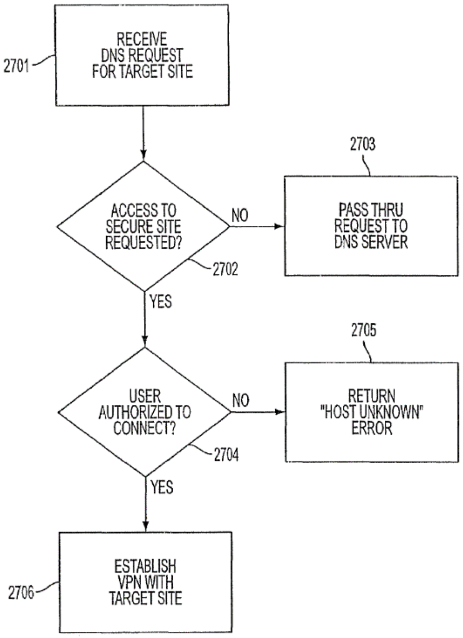

2701
RECEIVE DNS REQUEST FOR TARGET SITE
ACCESS TO SECURE SITE REQUESTED?
2702
NO
2703
PASS THRU REQUEST TO DNS SERVER
YES
2705
USER AUTHORIZED TO CONNECT?
2704
NO
RETURN "HOST UNKNOWN" ERROR
YES
2706
ESTABLISH VPN WITH TARGET SITE

FIG. 27

As shown in Figures 26 and 27, browser (2605) within user computer (2601) generates a DNS request for an IP address corresponding to a domain name of a target computer, such as secure target site 2604 and/or unsecure target site 2611.

Appx198 (37:60-38:43). Instead of conventional DNS server (2609) receiving the request, DNS proxy (2610) intercepts the request and determines whether it is for a secure website. Appx198 (37:60-62, 38:44-45). According to a preferred embodiment, the determination is made by checking the domain name against domain name tables. Appx198 (37:60-66).

If the domain name is listed therein, DNS proxy (2610) determines that the request is requesting access to a secure site, and may determine whether the user and/or computer 2601 is authorized to access the secure web site. Appx198 (37:60-66, 38:44-57). If so, DNS proxy (2610) automatically initiates a VPN between browser (2605) and secure target site (2604). Appx198 (37:62-38:2).

If the domain name is *not* listed in the domain name table, DNS proxy (2610) determines that the request is *not* seeking access to a secure site and forwards the request to conventional DNS (2609) without initiating a VPN. Appx198 (38:12-16, 38:40-43). This contrasts with a situation where, for example, an unauthorized user has requested lookup of a secure site, which would instead cause the DNS proxy to return a "host unknown" error to the user. Appx198 (38:16-19).

One embodiment for encrypting VPNs is the Tunneled Agile Routing Protocol ("TARP"). A client directly addresses a target using the IP address of the target. Appx183 (7:34-8:30). To avoid detection during routing, however, this target IP address "is concealed behind an outer layer of encryption generated using a link key." Appx183 (7:53-54). TARP routers sit between the client and target and "can use [a] link key to reveal the true destination of a TARP packet." Appx183 (7:34-62). Based on a "time to live counter" designed to "help foil traffic analysis," a TARP router "determine[s] … whether it should forward the TARP packet [] to another TARP router [] or to the destination TARP terminal." Appx183 (8:10-30).

Claim 1 of the '151 patent recites:

> A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:
>
> (i) determining whether the intercepted DNS request corresponds to a secure server;
>
> (ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and
>
> (iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

Appx202 (claim 1).

Claim 7 of the '151 patent recites:

> A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:
>
> (i) intercepting a DNS request sent by a client;
>
> (ii) determining whether the intercepted DNS request corresponds to a secure server;
>
> (iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

> (iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

Appx203 (claim 7).

Claims and 1 and 7 were determined to be unpatentable by the Board in another proceeding (IPR2015-01047), and this Court affirmed that finding. *See VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 2020-2271, 2023 WL 2708975, at *11 (Fed. Cir. Mar. 30, 2023). Claims 5 and 11 of the '151 patent (which include the features of claims 1 and 7, respectively) were not at issue in that proceeding. Claim 5 recites:

> The data processing device of claim 1, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

Appx203 (claim 5).

Claim 11 recites:

> The computer readable medium of claim 7, wherein automatically initiating the encrypted channel between the client and the secure sewer comprises establishing an IP address hopping scheme between the client and the secure server.

Appx203 (claim 11).

Claim 13 is another independent claim, also found unpatentable in *Mangrove*. *See* 2023 WL 2708975, at *11. Claim 13 recites:

> A computer readable medium storing a domain name server (DNS) module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

> (i) determining whether a DNS request sent by a client corresponds to a secure server;
>
> (ii) when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and
>
> (iii) when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel between the client and the secure server.

Appx203 (claim 13).

III. THE ASSERTED REFERENCES

A. Kiuchi

Kiuchi describes a system, "closed HTTP" ("C-HTTP"), that allows a "user agent" to securely access resources on an "origin server" in a "closed" network, such as a network connecting hospitals. Appx3426. In Kiuchi, the user agent and origin server do not communicate directly, but instead communicate with proxies that sit between the user agent and origin server, which in turn "communicate with each other using a secure, encrypted protocol" called C-HTTP. *Id.*

Kiuchi's "proxy-proxy" system involves five different entities. A user agent requests information, such as patient data, residing in an "origin server." Appx3426-3429. A "client-side proxy" sits between the user agent and the Internet, while a "server-side proxy" sits between the Internet and the origin server. Appx3426. Finally, a "C-HTTP name server" facilitates exchange of necessary encryption keys between the client-side and server-side proxy. Appx3426-3427.

Communications in Kiuchi involves multiple separate connections, using two different protocols. The user agent and client-side proxy communicate with each other using the conventional "HTTP/1.0" protocol, as do the server-side proxy and the origin server. Appx3426. The client-side proxy and server-side proxy "communicate with each other using a secure, encrypted protocol" called "C-HTTP." *Id.* (abstract) (italics omitted). The client-side and server-side proxies communicate with the "C-HTTP name server" in C-HTTP protocol as well. Appx3430. Kiuchi makes clear that secure "C-HTTP-based communication is performed *only* between two types of C-HTTP proxies and between a C-HTTP proxy and C-HTTP name server"—not between the user agent and the origin server. Appx3430 (emphasis added). The proxies "*do not* communicate directly" with the user agent or origin server using C-HTTP. *Id.* (emphasis added); *see also* Appx3429-3430.

Communications in Kiuchi involve nine steps. *See* Appx3427-3430. Step (1) is "[c]onnection of a client to a client-side proxy." Appx3427. A user requests a resource using a URL. The user agent sends the request to the client-side proxy, including the URL and a host name (and a connection ID if a C-HTTP session has already been established). Appx3427.

In step (2), the "client-side proxy asks the C-HTTP name server whether it can communicate with the host"—*i.e.*, the origin server—"specified in a given URL." Appx3427. The C-HTTP name server determines whether the server-side proxy associated with the origin server is registered in the closed network and permitted to accept the connection from the client-side proxy; if so, the C-HTTP name server returns the server-side proxy's IP address (not that of the origin server). If the connection is not permitted, the C-HTTP name server "sends a status code which indicates an error," causing the client-side proxy to "perform[] DNS lookup, behaving like an ordinary HTTP/1.0 proxy." Appx3427.

In step (3), the "client-side proxy sends a request for connection to the server-side proxy," which is encrypted. Appx3427. Then in step (4), "[w]hen a server-side proxy accepts a request for connection from a client-side proxy, it asks the C-HTTP name server whether the client-side proxy is an appropriate member of the closed network." Appx3427-3428. If so, the name server returns the client-side proxy's IP address. Appx3428.

In step (5), the server-side proxy then generates a connection ID and sends it with other information to the client-side proxy. Appx3428. "When the client-side proxy accepts and checks [the information from the server-side proxy], the connection is established." Appx3428.

In step (6), the "client-side proxy forwards HTTP/1.0 requests from the user agent in encrypted form using C-HTTP format" to the server-side proxy, since the C-HTTP name server previously provided the IP address of the server-side proxy for purposes of communication. Appx3428.

In step (7), "[u]sing HTTP/1.0, [the] server-side proxy communicates with [the] origin server inside the firewall" by forwarding requests to the origin server. Appx3428. But "[f]rom the view of the user agent or client-side proxy, all resources appear to be located in [the] server-side proxy" (since only its address is provided by the C-HTTP name server). Appx3428.

In step (8), the origin server sends an HTTP/1.0 response, which the server-side proxy then encrypts in C-HTTP format and forwards to the client-side proxy. "Then, in the client-side proxy, the C-HTTP response is decrypted and the HTTP/1.0 response extracted" and sent to the client. Appx3428. Figure b shows an HTML document from the origin server "rewritten and forwarded" by the client-side proxy to the user agent. Appx3428.

In step (9), "[a] client-side proxy can send a request for closing the connection." Appx3429. This nine-step process is repeated for each request made by the user agent. *Id.*

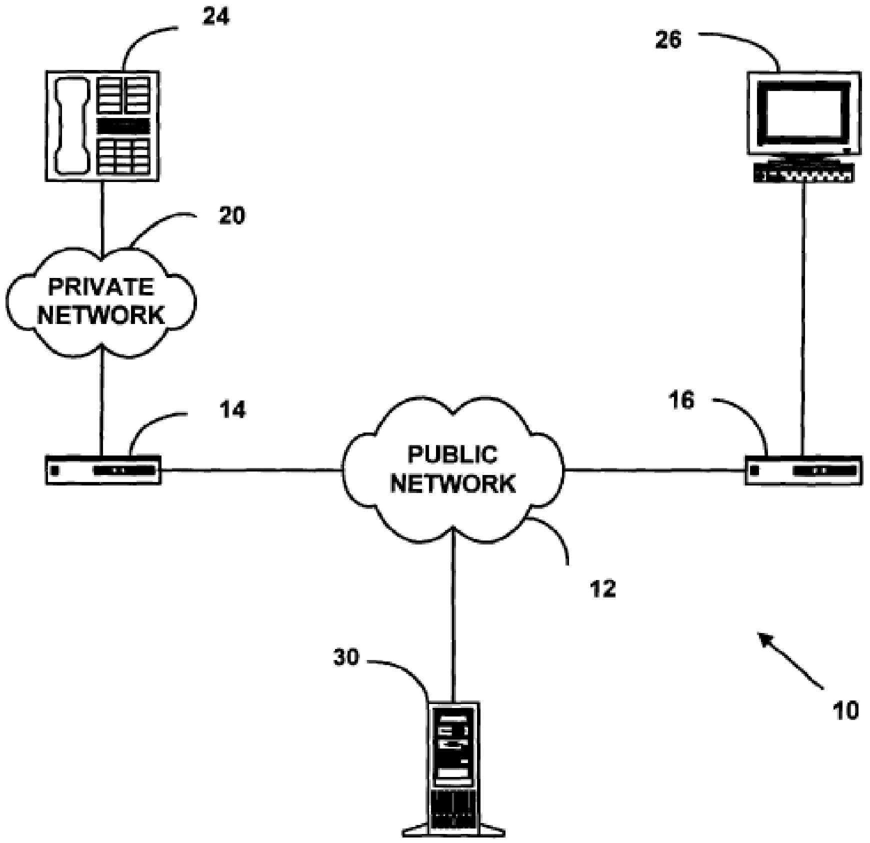
Kiuchi explained that its "proxy-proxy" design is "fundamentally different" from protocols that seek to "assure 'end-to-end' security" for direct communication between a user agent and origin server, Appx3429-3430, and has resulting benefits. First, because encrypted communications only occurs between proxies, "HTTP/1.0 compatible servers and clients can be used as they are," without modification to support security. Appx3430. Second, the multiplicity of connections in the proxy-

proxy approach makes eavesdropping difficult. An outsider would have to (a) "find the IP address and port number of a server-side proxy"; (b) "get the public key of the server-side proxy in order to send a valid C-HTTP request for C-HTTP connection"; (c) "make a TCP connection to a target server-side proxy using a certain client-side proxy's IP address"; and (d) obtain the private key and C-HTTP hostname of the client-side proxy to prove its legitimacy to the server-side proxy. Appx3430.

B. Beser

Beser "relates to communications in data networks," Appx4719 (1:8-9), and recognizes that "the Internet is not a very secure network," Appx4719 (1:26-27). Prior art methods attempted to secure communications by "encrypt[ing] the information inside the IP packets before transmission." Appx4719 (1:54-56). Beser teaches that this method is not secure because a determined hacker could accumulate enough packets from a source to decrypt the message, Appx4719 (1:56-58), and also imposes significant computational burden, especially in the context of voice and audio data, Appx4719 (1:58-67; 2:8-17).

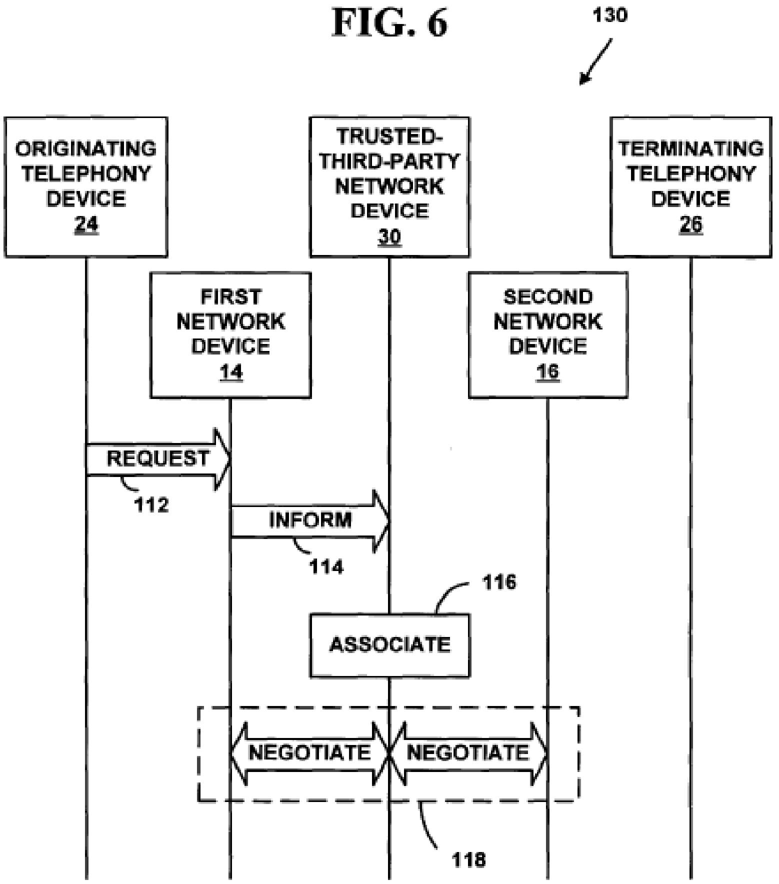
Beser's solution involves "initiating a tunneling association between an originating end [24] and a terminating end [26]" facilitated by an intermediary, trusted-third-party network device 30. Appx4719 (2:45-67); Appx4722 (7:62-64). Figure 1 illustrates this solution:



Appx4702 (Fig. 1).

When an originating device 24 wants to communicate with a terminating device 26, it sends a tunnel initiation request 112 to first network device 14. Appx4722 (7:65-66). This request "includes a unique identifier for the terminating end of the tunneling association." Appx4722 (8:1-3).

FIG. 6



Appx4707 (Fig. 6).

The first network device 14 then sends an "inform" message (step 114) with the tunnel initiation request (step 112) to trusted-third-party network device 30 via IP packets 58. Appx4724 (11:9-12). Device 30 associates a public IP address of a second network device 16 with the unique identifier of terminating device 26. Appx4722 (8:4-7); Appx4724 (11:26-32). The first and second network devices 14 and 16 then "negotiate" private IP addresses through the public network 12. Appx4722 (8:9-15); Appx4724 (11:58). This "negotiation" assigns a first and second private network address to the originating device 24 and the terminating device 26, respectively. Appx4724 (12:2-4).

Once assigned, the private network address of originating device 24 and the public IP address of first network device 14 are communicated to the second network device 16. Appx4725 (13:33-48). Similarly, the private network address of the terminating device 26 and the public IP address of the second network device 16 are communicated to the first network device 14. Appx4725 (14:19-33).

C. Kent

Kent is a Request for Comments document (RFC 2401) directed to IPsec, a type of security protocol. Appx4738. Among other things, Kent describes certain "[a]lgorithms for authentication and encryption. Appx4738.

D. Aventail Connect v3.01 and AutoSocks[3]

Aventail v3.01 ("Aventail") is an administrator's guide for configuring Aventail Connect, a client component of the Aventail ExtraNet Center, an extranet solution. Appx4554; Appx4558. Aventail Connect works in connection with extranet servers running the SOCKS protocol, including the Aventail ExtraNet

[3] As the Board noted, "[t]here is no dispute that the disclosure of AutoSOCKS is 'substantially similar' to Aventail Connect v3.01." Appx25. Given that both references were addressed in an identical manner below, VirnetX addresses them collectively here, too.

Server, the SOCKS 5 server component of the Aventail ExtraNet Center. Appx4558. Aventail discloses two primary embodiments:

(1) Aventail Connect may be used to provide secure *inbound* access, i.e., allowing an organization to provide its mobile employees and partners secure access to the organization's private network, extranet, or LAN from remote locations over the Internet. Appx4556; Appx4558; Appx4623.

(2) Aventail Connect may also be used as a simple proxy client for managed outbound access, e.g., from a corporate network to the Internet, through a SOCKS-compliant server. Appx4556; Appx4558; Appx4610-4612.

In the first embodiment, Aventail Connect accesses the private network through the Aventail ExtraNet Server. Appx4623. The Aventail ExtraNet Server restricts inbound access by allowing only authorized client computers running Aventail Connect to send or receive data to or from a computer on the private network, and provides an encrypted connection between the Aventail ExtraNet Server and the external client computer. Appx4614.

In the second embodiment, Aventail Connect may be configured to route certain traffic from a client computer running Aventail Connect to a SOCKS-compliant proxy server to traverse a firewall, Appx4557-4558, or in some cases, to traverse multiple firewalls using successive proxy servers, Appx4610-4615. Routing is accomplished, in part, by an administrator first defining which of several possible SOCKS proxy servers Aventail Connect should use when routing connections. Appx4584-4586 (figure depicting that a user may choose SOCKS v4, SOCKS v5, or HTTP proxy). The administrator may then define destinations (e.g., hostnames) and create redirection rules. Appx4586-4588. A redirection rule defines, for a particular destination, what type of traffic (i.e., TCP and/or UDP) will be allowed to be routed to that destination, and which proxy server will be used to route that traffic. Appx4589-4591.

E. Aziz

Aziz discloses a system "for dynamically configuring authorized clients with the address of a protected host and the key and address of an intermediate device (e.g., encrypting firewall, encrypting router, secure gateway) which is protecting a number of hosts on a private network" behind the intermediate device. Appx3489 (4:3-9).

Aziz explains that "outside NS" 120 may receive a query for a host address located within domain 100 and may check its database for an SX record the requested host name. Appx3492 (9:29-53). An SX record is a resource record that "contains the identifier (e.g., name or address) of a 'secure exchanger,'" such as firewall 110. Appx3490 (6:23-40). If an SX record exists, then outside NS 120 may include the SX record in the response to the requester, which may also include the requested host address, if available. Appx3492 (9:54-10:5).

Aziz also discloses a resolver 225, which is included in the "authorized client" 210, Appx3491 (8:5-50); Appx3477-3479 (Figs. 2A-2C), and that receives a response to the query for a host address. Appx3492 (10:39-41). If the response includes an SX record and the requested host address, then resolver 225 creates a tunnel map entry that provides the information "authorized client" 210 needs to encrypt messages to "inside host" 140. Appx3493 (11:13-60). Resolver 225 then returns the requested host address to an application 215, also located in "authorized client" 210. Appx3490 (5:23-28); Appx3493 (11:55-60). According to Aziz, "[t]his completes the execution" of the configuration process. Appx3493 (11:60-62).

F. Edwards

Edwards discloses a "secure object gateway . . . to give fine grain access control" to services located on the back end of a web server. Appx3503. With regard to Figure 4, Edwards discloses that a naming interceptor at an object gateway intercepts a request to resolve a name corresponding to one of the services that is accessible from the back-end of the web server. *Id.* The naming interceptor returns a reference to a service interceptor also located at the object gateway. *Id.* A plugin at the requesting web server then invokes the service interceptor to reach the backend service. *Id.*

Edwards discloses that the plugin invoking the services in this manner may first need to be authenticated and authorized. *Id.* Edwards also discloses, however, that an administrator may remove the authentication and/or authorization requirements for accessing these services (Edwards also refers to the services as "available targets"). Appx3504.

IV. PROCEDURAL BACKGROUND

A. The PTO's Merger of the *Cisco* and *Apple* Reexaminations

Apple initiated *inter partes* reexamination no. 95/001,697 ("the 697 reexamination") in July 2011. Shortly thereafter, in August 2011, Cisco initiated the proceeding below ("the 714 reexamination"). Apple and Cisco raised distinct grounds in each proceeding. In the 697 reexamination, Apple proposed rejections based on Aventail Connect v3.01; Aventail AutoSOCKS; BinGO; the combination of Beser and Kent; Wang; and the combination of Wang and Beser. In the 714 reexamination (at issue here), Cisco proposed rejections based on Kiuchi; Wesinger; Blum; and Aziz.

On March 15, 2012, the PTO *sua sponte* merged the two reexaminations. In doing so, however, the PTO specified that the rejections proposed in each proceeding should be treated separately—in other words, that the rejections Apple proposed in the 697 reexamination proceeding should *not* be treated as if they had been proposed by Cisco in the 714 reexamination proceeding, and vice versa. The PTO expressly recognized that "each inter partes reexamination requester's appeal *must only be taken* from the finding(s) of patentability of claims in the RAN that the individual third party requester proposed in the request, and any that the individual third party requester properly added during the examination stage of the merged proceeding." Appx326-327 (emphasis added).

On March 23, 2015, VirnetX filed a petition to sever the two reexaminations and to terminate the 697 reexamination pursuant to the estoppel provision of pre-AIA 35 U.S.C. § 317(b). Specifically, VirnetX sought termination of the 697 reexamination at least with respect to claims 1-6 and 13-16 of the '151 patent—claims the validity of which have been upheld in district-court litigation (claims 1 and 13) and claims dependent from them. VirnetX argued that section 317(b) precluded the PTO from maintaining the reexamination as to those claims because the requester, Apple, had lost an appeal of validity issues relating to those claims in a civil case. Appx925-927 (discussing *VirnetX, Inc. v. Cisco Sys., Inc. and Apple Inc.*, 767 F.3d 1308, 1324 (Fed. Cir. 2014)). The PTO, however, denied VirnetX's petition, disagreeing with VirnetX's interpretation that section 317(b) requires termination of a reexamination as soon as the Federal Circuit issues a final decision on validity. The PTO reasoned that section 317(b) did not apply because, while Apple had lost its validity challenge on appeal, the Federal Circuit had remanded on other issues. The PTO viewed the remote possibility that Apple might seek Supreme

Court review of the validity ruling in a petition for a writ of certiorari filed after the conclusion of the remand proceedings and any subsequent appeal sufficient to foreclose application of Section 317(b). Appx989-991.

As a result, the PTO continued prosecution in the merged Apple and Cisco reexamination proceedings. On February 24, 2016—almost a year after VirnetX requested severance—the Examiner issued an Action Closing Prosecution and a Right of Appeal Notice ("RAN") that adopted rejections from both the 697 reexamination and the 714 reexamination proceedings. Appx69. VirnetX appealed the adopted rejections on March 24, 2016 (and Cisco cross-appealed the non-adoption of certain rejections), and the Examiner issued an Examiner's Answer, which maintained the RAN. Appx69.

B. The PTO's Severance, and Partial Termination, of the *Apple* Reexamination

On June 2, 2017, VirnetX renewed its petition to sever the two reexaminations and to terminate the 697 reexamination in light of this Court's decision in *Fairchild (Taiwan) Corp. v. Power Integrations, Inc.*, 854 F.3d 1364 (Fed. Cir. 2017). Appx2130-2147. *Fairchild* involved parallel *inter partes* reexamination and infringement litigation. In the infringement action, the district court rejected the patent challenger's invalidity defense; this Court affirmed that ruling, while remanding for further proceedings on other issues. 854 F.3d at 1365. The Court then held that the district court's validity determination constituted a "final decision" under section 317(b), as it had affirmed that validity determination and the time to petition for a writ of certiorari had expired—even though proceedings on other issues would continue on remand. *Id.* at 1365-66. As the Court explained, "[b]y its terms, § 317(b) is concerned with a final decision 'that the party has not sustained its burden of proving the invalidity of any patent claim,'" and there was no "reason to believe, that any unresolved issue on remand would have any effect on the now-final … patent validity determinations." *Id.* at 1366.

On August 1, 2019, this Court issued a decision on appeal from a different set of reexaminations involving VirnetX and Apple, *VirnetX Inc. v. Apple Inc.*, 931 F.3d 1363 (Fed. Cir. 2019). That decision confirmed that, under *Fairchild*, affirmance of a district court's validity decision triggers section 317(b) estoppel even if the court of appeals remands on other issues that might result in a later appeal and subsequent certiorari petition. The Court explained that *Fairchild* "controls" and that affirmance of the district court judgment against Apple on the issue of validity constitutes "'a final decision' under § 317(b) once the time to file a certiorari petition has passed." *VirnetX*, 931 F.3d at 1371. The Court further explained that "the statutory text and purpose of § 317(b) compels [the] reject[ion of a] theory that the entire case must be resolved"—including remand proceedings on non-validity issues. *Id.* at 1374.

On October 16, 2019, in light of these decisions, the PTO granted VirnetX's petition. Appx2192-2210. The PTO severed the 697 reexamination proceeding from the 714 reexamination proceeding. Appx2197. The PTO then acknowledged that, in light of section 317(b)'s mandate that "an inter partes reexamination requested by [an estopped party] on the basis of [estopped] issues may not thereafter be maintained by the Office," Appx2199, "[a]ny rejection which is presently applied against claims 1-6 and 13-16 of the '151 patent, i.e., the patent under reexamination, in the '1697 reexamination proceeding *will not be further maintained by the Office*," Appx2209 (emphasis in original).

The PTO did not, however, issue a new RAN that omitted the estopped Apple-proposed rejections. Instead, on November 6, 2019, the Office mailed an appeal docketing notice based on the old RAN, which contained rejections from both the 697 reexamination and the 714 reexamination proceedings—including the Apple-proposed rejections that could no longer be maintained under section 317(b). Appx2217-2219. In light of that deficiency, on November 12, 2019, VirnetX requested issuance of a new RAN. The PTO and the Board never ruled on VirnetX's request.

C. The Board's Decisions Below

On June 23, 2020, the Board issued a decision ("June 23, 2020 Board Decision") affirming rejections that were proposed and adopted as part of Apple's 697 reexamination proceeding against such claims, including rejections that Apple—but not Cisco—had proposed based on Aventail Connect v3.01; Aventail

AutoSOCKS; and the combination of Beser and Kent. Appx62. In a footnote, the Board stated that, despite the PTO's prior ruling that rejections "against claims 1-6 and 13-16 [of the '151 patent] will not be further maintained by the Office ... in the '1697 reexamination proceeding," the Board "considers rejections maintained by the Office against claims 1-6 and 13-16 for rejections as part of Reexamination Control No. 95/001,714 *even though certain rejections originated from Reexamination Control No. 95/001,697*." Appx3 n.1 (emphasis added).

On June 29, 2020, VirnetX filed a petition seeking vacatur of the Board's June 23, 2020 decision. VirnetX argued that the Board's decision to adopt the rejections proposed by Apple in the 697 reexamination proceeding was contrary both to the PTO's October 16, 2019, which severed the 697 reexamination and terminated it as to the estopped claims, as well as to this Court's *Fairchild* and *VirnetX* decisions.Appx2224-2227. Specifically, VirnetX argued (Appx2224-2225) that—as the PTO recognized—section 317(b) precluded the estopped rejections from being "further maintained by the Office." 35 U.S.C. § 317(b) (2006). VirnetX also explained that the Board's June 23, 2020 Decision could not be upheld on the basis that the RAN adopting both sets of rejections was issued prior to the proceedings' severance in October 2019. VirnetX originally requested the severance on March 23, 2015, prior to the RAN's issuance, and the Board's denial of that request was based on a misreading of section 317(b)—a misreading that this Court subsequently corrected. Appx2226-2227. Had the PTO properly granted VirnetX's petition at the outset, the rejections-at-issue would never had become part of the Examiner's RAN and would not have been before the Board. Appx2226-2227.

On September 28, 2020, the Board denied VirnetX's petition to vacate. The Board reasoned that the PTO's decision that finally severed the two reexamination proceedings only ordered termination of the reexamination of the estopped Apple-proposed rejections in the 697 proceeding (initiated by Apple), but not in the 714 proceeding (brought by Cisco). Appx75-77. The Board also opined that "the Examiner is not limited only to the prior art or rationales proposed by a requesting party, in this case Cisco," but can "maintain rejections over the best references at his or her command." Appx77-79. On that basis, the Board concluded that section 317(b) "does not require withdrawal of any specific rejections maintained by the Examiner in the '1714 reexamination filed by Cisco." Appx79.

On October 13, 2020, VirnetX sought rehearing of the Board's decision, including rehearing by the Board's Precedential Opinion Panel—the Board's supervisory body at that time that reviewed cases involving significant policy or procedural issues. Appx2252-2265. The Precedential Opinion Panel denied the request for review, Appx81-82, and the Board then denied rehearing on April 27, 2021, Appx83-104.

The Board, however, remanded the 714 reexamination to the Examiner in response to VirnetX's request to reopen prosecution, so that the Examiner could address the Board's new grounds of rejection with respect to claims 6 and 12. Appx2272-2275. The Examiner issued a decision with respect to these new grounds on August 17, 2021, which the Board affirmed on March 24, 2022. Appx105-117. VirnetX sought rehearing of the Board's decision, reiterating its position that the Board's consideration of the Apple-initiated rejections was improper. *See* Appx2302-2308 & n.1. The Board denied rehearing on February 10, 2023, refusing to "revisit [VirnetX's] continued objections to the application of certain rejections originating from reexaminations originally requested by Apple, Inc." Appx118-135.

D. This Court's Decision on a Motion to Remand

VirnetX timely appealed to this Court on April 7, 2023, and the Court docketed the appeal on April 20, 2023. *See* Docket No. 1. On April 21, 2023, Cisco filed a cross-appeal, which the Court docketed on April 27, 2023. *See* Appeal No. 23-1809, Docket No. 1. The Court consolidated both appeals. *See* Docket No. 3.

On September 29, 2023, VirnetX filed a motion asking this Court to vacate the Board's decision below and remand these appeals to the Board for consideration of Cisco's patentability challenges without the improper reliance on Apple-proposed rejections. *See* Docket No. 19 at 1-16.

In its motion, VirnetX noted that, subsequent to the Board's denial of rehearing, this Court in *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 20-2271, 2023 WL 2708975 (Fed. Cir. Mar. 30, 2023), affirmed the Board's invalidity finding as to several claims of the '151 patent, including claims 1-2, 7-8, and 13-14 that are at issue in this appeal. Docket No. 19 at 16 n.3. Thus, VirnetX observed, *Mangrove* at present renders moot the patentability of these claims, since they are now slated for cancellation. *Id.*[4]

On December 27, 2023, the Court entered an order "deem[ing] it the better course" to have the parties "address[] the issue in their merits briefs." Docket No. 24 at 2. The Court therefore denied VirnetX's motion to remand "without prejudice to the parties including their arguments in their merits briefs." *Id.*

SUMMARY OF ARGUMENT

1. Pre-AIA section 317(b) precludes the Board from maintaining an *inter partes* reexamination after "a final decision" that the patent challenger failed to prove invalidity of the challenged claims in a civil case. *See* 35 U.S.C. § 317(b) (2006). Despite this statutory command, the Board in this reexamination used the Apple-proposed rejections—which Cisco never asserted in this proceeding—to invalidate the very claims as to which the PTO has terminated the reexamination in

[4] VirnetX also noted that it has sought certiorari from this Court's decision in *Mangrove* in *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 23-315 (U.S. docketed Sept. 27, 2023). *See* Docket No. 19 at 16 n.1.

the Apple-initiated proceeding. The Board cannot evade the mandate of section 317(b)'s estoppel provision by importing *the very same rejections* advanced in a reexamination proceeding it is expressly prohibited from "maintaining" into a different reexamination proceeding, and then using them to invalidate the claims whose validity this Court has already affirmed. The Board's decision represents an impermissible end-run around section 317(b) and defies this Court's decisions in *Fairchild (Taiwan) Corp. v. Power Integrations, Inc.*, 854 F.3d 1364 (Fed. Cir. 2017), and *VirnetX Inc. v. Apple Inc.*, 931 F.3d 1363 (Fed. Cir. 2019).

2. The Board's decision below addressed claims that have since been slated for cancellation as a result of this Court's decision in another appeal involving the '151 patent, *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 20-2271, 2023 WL 2708975 (Fed. Cir. Mar. 30, 2023). The Board's decision with respect to those claims (claims 1-2, 6-8, and 12-14) is now moot. When mootness is caused by a prior court decision preventing review, vacatur is required because judicial review of the decision with respect to the moot claims "was prevented through happenstance." *United States v. Munsingwear, Inc.*, 340 U.S. 36, 39-41 (1950); *see also Apple Inc. v. Voip-Pal.com, Inc.*, 976 F.3d 1316, 1321 (Fed. Cir. 2020). Under traditional principles of vacatur, the Board's decision with respect to those claims should be vacated.

3. Regardless, Cisco's (and Apple's) underlying unpatentability theories include critical defects, and the Board's decision to adopt them should be reversed on the merits. With respect to Kiuchi, the Board improperly relied on a materially new mapping of Kiuchi to the claimed DNS proxy module that incorrectly combined two separate components that are not part of the same module. And in addressing the requirement in the claims that "when [the / the intercepted] DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer," Appx202-203 (claims 1, 7, 13), the Board pointed vaguely to Kiuchi's disclosure of a "DNS lookup," Appx12. But the Board overlooked that Kiuchi never says what it sends for DNS lookup, much less that the client-side proxy forwards the request that it receives.

With respect to Beser, the Board's analysis of the "forwarding" limitation was even more egregious. The Examiner relied on a failed inherency theory for why Beser met the limitation, and when VirnetX explained why, Appx1617-1618, the Board simply asserted, without explanation, that "Patent Owner does not meaningfully address the Examiner's rationale." Appx44.

For Aventail (and AutoSOCKS), the Board gave inadequate consideration to VirnetX's argument that the claimed "secure channel" should be construed as a "direct channel that is secure." The Board asserted that "[e]ven if the claims of the '151 Patent do require a 'direct' channel, Patent Owner has not sufficiently explained how such a requirement in the claims necessarily distinguishes Aventail Connect v3.01." Appx22. Yet this Court previously analyzed Aventail and described it as an example of an indirect connection. *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, 778 F. App'x 897, 910 (Fed. Cir. 2019).

Finally, the Board incorrectly determined that Aziz disclosed the claimed "determining whether [the intercepted / a] DNS request corresponds to a secure server." Appx202-203 (claims 1, 7, 13). The Board pointed to a teaching in Aziz where a data field called an "SX record", which can correspond to a "secure exchanger," is returned in response to a name server request. But Aziz makes no mention of using this information to determine whether a DNS request corresponds to a secure server.

ARGUMENT

I. STANDARD OF REVIEW

This Court reviews the Board's legal conclusions de novo and its factual findings for substantial evidence. *EmeraChem Holdings, LLC v. Volkswagen Grp. of Am., Inc.*, 859 F.3d 1341, 1345 (Fed. Cir. 2017). Anticipation is a fact question reviewed for substantial evidence. *Power Integrations, Inc. v. Lee*, 797 F.3d 1318, 1323 (Fed. Cir. 2015). Factual findings underlying obviousness are also reviewed for substantial evidence, while the ultimate question of obviousness is reviewed de novo. *Dell Inc. v. Acceleron, LLC*, 818 F.3d 1293, 1298 (Fed. Cir. 2016). "[T]he

Board's procedures [are reviewed] for compliance with the Administrative Procedure Act ('APA') de novo." *EmeraChem*, 859 F.3d at 1345.

II. THIS COURT SHOULD VACATE THE BOARD'S DECISION AND REMAND FOR RECONSIDERATION WITHOUT THE IMPROPER APPLE-INITIATED REJECTIONS

This Court should vacate the Board's decision to maintain in the 714 reexamination Apple-initiated rejections from the 697 reexamination that the PTO was statutorily required to terminate. That decision is flatly contrary to the unambiguous command of section 317(b). In relevant part, pre-AIA Section 317(b) unambiguously mandates:

> ***Once a final decision has been entered*** *against a party* in a civil action arising in whole or in part under section 1338 of title 28, *that the party has not sustained its burden of proving the invalidity of any patent claim in suit* ..., then neither that party nor its privies may thereafter request an inter partes reexamination of any such patent claim on the basis of issues which that party or its privies raised or could have raised in such civil action or inter partes reexamination proceeding, and *an inter partes reexamination requested by that party or its privies on the basis of such issues* ***may not thereafter be maintained by the Office***, notwithstanding any other provision of this chapter.

35 U.S.C. § 317(b) (2006) (emphasis added).

As this Court held, section 317(b) prevents a party from seeking reexamination of claims once it loses a validity challenge to those claims in district court and this Court affirms. *Fairchild*, 854 F.3d at 1365-66; *VirnetX*, 931 F.3d at 1371-72. Section 317(b) thus was triggered when this Court rejected Apple's district-court validity challenge to claims of the '151 patent on September 16, 2014.

See VirnetX, Inc. v. Cisco Sys., Inc. and Apple Inc., 767 F.3d 1308 (Fed. Cir. 2014). At that point, section 317(b) prohibited Apple from "request[ing]"—and the PTO from "maintain[ing]"—a reexamination "on the basis of [the] issues" Apple had lost in the district-court litigation.

Under the correct understanding of section 317(b), the PTO should have terminated Apple's reexamination request in the 697 reexamination when VirnetX petitioned for termination (and severance) on March 23, 2015. The Board also should have declined to further consider—much less adopt—the rejections Apple proposed in the 697 reexamination. Cisco had *not* proposed any of those same rejections in the separately filed 714 reexamination, nor did the Director identify these rejections or the prior art asserted by Apple when he determined that Cisco's request raised a "substantial new question of patentability." Appx234; Appx239-241. And the plain language of section 317(b) prohibited the PTO from "maintain[ing]" a reexamination "on the basis of [the] issues" Apple had proposed. 35 U.S.C. § 317(b) (2006).

The Board, however, asserted that the Examiner could consider the Apple-proposed rejections as the basis for his RAN because, at the time of its issuance on February 24, 2016, the 697 and the 714 reexaminations were still merged, and the PTO's October 16, 2019 Petition Decision—which terminated the reexamination of those rejections in the *697 reexamination*—purportedly permitted the Board to consider those same rejections in the ongoing *714 reexamination*. Appx77-78. But the only reason the Examiner continued to consider Apple's estopped rejections when preparing the RAN was because the PTO erroneously failed to terminate the reexamination of those rejections in response to VirnetX's petition (which was filed nearly a year before the RAN's issuance). The PTO cannot erroneously reject VirnetX's invocation of section 317(b)'s estoppel provision by refusing to terminate the 697 proceeding, and then rely on that erroneous decision to effectively continue the reexamination of the affirmed claims that section 317(b) plainly forbids. That is an impermissible end-run around section 317(b).

When Congress prescribes a specific timing sequence, that choice "is significant in construing statutes." *United States v. Wilson*, 503 U.S. 329, 333 (1992) (citing cases). Here, Congress prescribed that section 317(b)'s estoppel should apply "[o]nce" there is a final affirmed district court validity decision; from that point onward, a reexamination of those affirmed claims "may not thereafter be maintained by the Office." 35 U.S.C. § 317(b) (2006). Congress did not provide that this affirmative directive to the agency may be ignored so long as the PTO persists in an erroneous interpretation of its governing statute, only to be triggered "once" this Court instructs the PTO on the correct application of its statutory mandate.

For that reason, the Board's insistence that the October 16, 2019 Petition Decision "only ordered a new RAN be mailed in the '1697 reexamination proceeding, [and] not in the ’1714 reexamination proceeding,” Appx78, is beside the point. In the October 16, 2019 Petition Decision, the PTO correctly recognized that section 317(b) prohibited maintaining reexamination of claims 1-6 and 13-16 of the ’151 patent based on the Apple-proposed rejections. Appx2192-2210; *supra* at 24-25. The Board’s September 28, 2020 decision is rooted in the faulty premise that the merged reexamination proceedings were properly maintained until they were finally severed in October 2019, and that the Board, in an appeal from the Examiner’s decision in the 714 reexamination, can continue to consider the estopped rejections proposed in the 697 reexamination. But if the PTO had properly granted VirnetX’s original petition to sever the two proceedings and to terminate Apple’s proceeding with respect to the claims subject to section 317(b)’s estoppel, the estopped rejections would have never been part of the Examiner’s RAN, nor remained before the Board.[5]

Finally, the September 28, 2020 Decision invokes the Examiner’s authority to

[5] For that reason, the Board’s reliance on *In re Affinity Labs of Texas, LLC*, 856 F.3d 883 (Fed. Cir. 2017), *see* Appx76-77, is misplaced. *Affinity Labs* held that section 317(b)’s estoppel applies only to the “requester that was a party to the civil action or its privies”—but not to other, unrelated parties. 856 F.3d at 893. Here, *Affinity Labs* would have required termination of the reexamination requested by Apple with respect to the upheld claims as of September 2014—the date on which this Court affirmed the district court’s validity ruling. Nothing in *Affinity Labs* permits the PTO to continue to consider those rejections simply because the estopped reexamination proceeding is merged with another proceeding where a different requester proposed different rejections.

consider "'the best references at his or her command,'" including "'patents and printed publications … cited by another reexamination requester.'" Appx78-79 (quoting 37 C.F.R. § 1.937, MPEP § 2656(A)). As an initial matter, these regulatory provisions cannot override the plain statutory command of section 317(b). Moreover, these provisions simply explain that the Examiner is not limited to the prior art asserted by the requester, but may consider the "patents and printed publications" that may have been cited elsewhere. They do not authorize the Examiner to rely on rejections, arguments, and evidence from the proceeding that is *terminated* by operation of section 317(b). That is particularly so in this proceeding, where Apple introduced extensive briefing and evidence in the merged reexamination proceedings well after VirnetX's petition to sever and terminate. *See* Appx2226 & n.4. None of Apple's arguments or evidence should have been part of those proceedings because VirnetX's petition to sever/terminate should have been granted at the outset. To treat these provisions as somehow permitting the Examiner to continue considering those rejections, arguments, and evidence in the Cisco-initiated 714 proceeding is to render section 317(b) a nullity.

Furthermore, under pre-AIA sections 312 and 313 applicable to this reexamination proceeding, the Board's consideration is limited to the resolution of the "substantial new question of patentability" identified by the Director when instituting review. 35 U.S.C. § 312(a) (2006); *see also id.* § 313 (2006).[6] Thus, pre-AIA section 312 provides that, in deciding whether to institute reexamination, "the Director shall determine whether a substantial new question of patentability affecting any claim of the patent concerned *is raised by the request*." 35 U.S.C. § 312(a) (2006) (emphasis added). And pre-AIA section 313 provides, in turn, that "[i]f, in a determination made under section 312(a), the Director finds that a substantial new question of patentability affecting a claim of a patent is raised, the determination shall include an order for inter partes reexamination of the patent *for resolution of the question*." 35 U.S.C. § 313 (2006) (emphasis added). Here, the Director's order granting Cisco's request in the 714 reexamination proceeding did not refer to Apple's proposed rejections or the prior art proposed by Apple; the order identified "a substantial new question of patentability" based solely on the prior art and rejections advanced by Cisco. *See* Appx234; Appx239-241.

The only reason the rejections from the 697 reexamination were before the Board in the reexamination below is because of the PTO's erroneous denial of VirnetX's initial petition to sever/terminate and the subsequent delay in granting VirnetX relief under section 317(b). The Board Decision in effect uses the PTO's admittedly erroneous denial of VirnetX's original petition as a basis to consider and

[6] Both the 697 reexamination and the 714 reexamination were filed prior to September 16, 2011.

rely upon rejections the Office is forbidden from maintaining. That has no basis in the law, and warrants vacatur and remand by this Court.

III. THE BOARD'S FINDINGS AS TO CLAIMS THAT ARE SLATED FOR CANCELLATION MUST BE VACATED AS MOOT

The Board below found that claims 1-2, 5-8, and 11-14 of the '151 patent are unpatentable. Appx62. Except for claims 5 and 11, all these claims have been found unpatentable in *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 20-2271, 2023 WL 2708975 (Fed. Cir. Mar. 30, 2023), and so are slated for cancellation. As a result, consideration of the unpatentability of these claims is moot, and the Board's decision with respect to those claims should be vacated. *See, e.g.*, *Eisai Co. v. Teva Pharms. USA, Inc.*, 564 U.S. 1001 (2011) (vacating the Federal Circuit's judgment where the case became moot while a petition for *en banc* rehearing was pending) (citing *United States v. Munsingwear, Inc.*, 340 U.S. 36 (1950)); *Stewart v. S. Ry. Co.*, 315 U.S. 784 (1942) (vacating the judgment that became moot on petition for rehearing after case was decided on the merits, 315 U.S. 283 (1942)); *Munsingwear*, 340 U.S. at 40 (vacatur is proper where "review of [a judgment] was prevented through happenstance").

"Vacatur is in order when mootness occurs through happenstance—circumstances not attributable to the parties—or … the unilateral action of the party who prevailed in the lower court." *Arizonans for Official English v. Arizona*, 520 U.S. 43, 71-72 (1997) (internal quotation marks and citation omitted). In

Munsingwear, the Supreme Court held that "[t]he established practice of the Court in dealing with a civil case from a court in the federal system which has become moot while on its way here or pending our decision on the merits is to reverse or vacate the judgment below and remand with a direction to dismiss." 340 U.S. at 39. Vacatur "clears the path for future relitigation of the issues between the parties and eliminates a judgment, review of which was prevented through happenstance." *Id.* at 40.

The Supreme Court subsequently clarified in *Bonner Mall* that "the reference to 'happenstance' in *Munsingwear* must be understood as an allusion to this equitable tradition of vacatur. A party who seeks review of the merits of an adverse ruling, but is frustrated by the vagaries of circumstance, ought not in fairness be forced to acquiesce in the judgment." *U.S. Bancorp Mortg. Co. v. Bonner Mall P'ship*, 513 U.S. 18, 25 (1994). Because of the equitable tradition of vacatur, when a party has *settled* a case while an appeal is pending, and thus has "forfeited his legal remedy by the ordinary processes of appeal or certiorari," vacatur is generally not proper. *Id.* But when a party has not forfeited its right to appeal, "'[w]here it appears upon appeal that the controversy has become entirely moot, it is the *duty* of the appellate court to set aside the decree below and to remand the cause with directions to dismiss.'" *Great W. Sugar Co. v. Nelson*, 442 U.S. 92 (1979) (quoting *Duke PowerCo. v. Greenwood County*, 299 U.S. 259, 267 (1936)).

This Court in *Apple Inc. v. Qualcomm* succinctly described the distinction between the circumstances where vacatur is inappropriate and those where it is required:

> *To one side* are cases in which an appellant, "frustrated by the vagaries of circumstance" or the "unilateral action" of the appellee, "ought not in fairness be forced to acquiesce in the judgment." *To the other* are cases like this one, in which "mootness results from *settlement*" such that "the losing party has voluntarily forfeited his legal remedy ... thereby surrendering his claim to the equitable remedy of vacatur."

17 F.4th 1131, 1137 (Fed. Cir. 2021) (quoting *Bonner* Mall, 513 U.S. at 24-25) (emphasis added). This case plainly does not fall on the side of cases "in which mootness results from settlement such that the losing party has voluntarily forfeited his legal remedy," but rather on the side where the "vagaries of circumstance"—another decision of this Court that issued due to parallel proceedings challenging the '151 patent—caused mootness. *Id.* (internal quotation marks and citation omitted).

Vacatur in this case would comport with the principles underlying the *Munsingwear-Bonner Mall* jurisprudence. As the Supreme Court explained, "[t]he principal condition" informing a court's decision whether to vacate the underlying decision on mootness grounds is "whether the party seeking relief from the judgment below caused the mootness by voluntary action." *Bonner Mall*, 513 U.S. at 24 (citing *United States* v. *Hamburg-Amerikanische Packetfahrt-Actien Gesellschaft*, 239 U.S. 466, 477-478 (1916); *South Spring Hill Gold Mining Co. v. Amador*

Medean Gold Mining Co., 145 U.S. 300, 302 (1892)). Here, no "voluntary action" by VirnetX is responsible for this Court's decision in *Mangrove*, which upheld the unpatentability of several of the claims at issue in this appeal. On the contrary, VirnetX argued strenuously that this Court should reverse the Board's unpatentability findings in that appeal. Thus, this is not a situation where VirnetX "surrender[ed its] claim to the equitable remedy of vacatur," and the judgment below is "simply unreviewed by [its] own choice." *Bonner Mall*, 513 U.S. at 25.

This Court and others have confirmed that, under the Supreme Court's precedent, vacatur is required where, as here, a prior decision has rendered an appeal moot. For instance, in *Apple Inc. v. Voip-Pal.com, Inc.*, the Court determined that "overlapping claims failed the Section 101 threshold" in a prior appeal. 976 F.3d 1316, 1321 (Fed. Cir. 2020). As a result, the Court vacated the Board's decisions as to those overlapping claims under *Munsingwear*. *Id.* Similarly, in *Mylan Pharmaceuticals Inc. v. Biogen Ma, Inc.*, this Court noted that, in a "companion case … this court found the [challenged] patent invalid for lack of written description," and, as a result, "agree[d] with the parties that th[e] appeal is moot and the underlying final written decision should be vacated." No. 2020-1673, 2022 WL 14461991, at *1 (Fed. Cir. Oct. 25, 2022) (describing *Voip-Pal.com* as "vacating Board decisions on patentability and remanding with instructions to dismiss IPRs as to those claims because a district court's invalidation of those claims rendered the appeal moot").

This Court is not alone in finding vacatur appropriate in such circumstances. For instance, in *NASD Dispute Resolution, Inc. v. Judicial Council of State of California*, the Ninth Circuit considered a case where "[the Ninth Circuit] and the California Supreme Court resolved the controversy with [prior] decisions in *Credit Suisse* and *Jevne*." 488 F.3d 1065, 1070 (9th Cir. 2007). As a result, the court of appeals explained, "[t]he present case is one in which 'happenstance,' not the parties' own actions, rendered the appeal moot." *Id.* And the Ninth Circuit confirmed that the "exception identified in *Bonner Mall* for *settlements* should not apply to judgments mooted by *court decisions* in other cases." *Id.* (emphasis added); *see also Panera, LLC v. Dobson*, 999 F.3d 1154, 1159 (8th Cir. 2021) ("Panera's action of filing an identical suit in the Delaware court one day after the district court's dismissal in Missouri does not satisfy the Supreme Court's narrow definition of voluntary action to justify invocation of the [*Bonner Mall*] exception to vacatur. Panera did not settle the case, nor did it fail to appeal.") (citing *NASD*, 488 F.3d at 1070) (internal quotation marks and additional citations omitted).

Indeed, this is the approach this Court has taken in an appeal involving another patent held by VirnetX (No. 2017-2594). While Appeal No. 2017-2594 was pending before the Court, all claims of the patent at issue in that appeal had been cancelled as a result of other Board proceedings, rendering that appeal moot. Both VirnetX and the PTO agreed that, under the *Munsingwear* line of cases, the Board's underlying decision had to be vacated. *See* Appeal No. 2017-2593 (lead appeal), Docket No. 65 at 2; *see also id.*, Docket No. 54 at 2; Docket No. 62 at 1. This Court agreed as well, vacating the Board's final decision at issue in Appeal No. 2017-2594 and remanding that case to the Board with instructions to dismiss. *See id.*, Docket No. 65 at 3. The same result should follow here with respect to the Board's decision as to claims of the '151 patent whose unpatentability is now moot.

IV. THE BOARD'S REJECTIONS ARE NOT ADEQUATELY EXPLAINED, AND ARE NOT SUPPORTED BY SUBSTANTIAL EVIDENCE

In any event, the Board's unpatentability findings are deficient. Rather than identify and apply a rejection that the Board thought was strongest, it attempted to show unpatentability through numerosity, applying rejections to the claims based on Kiuchi, Beser, Aventail/AutoSOCKS, and Aziz. The Board appears to have taken this approach to make up for the fact that each of the rejections has critical defects. As explained further below, each of the rejections should be reversed.

A. The Board's Kiuchi-Based Rejections Should Be Reversed

The Board found that "the combination of the 'client-side proxy' and the 'C-HTTP name server' disclosed in Kiuchi corresponds to the DNS proxy module that performs steps (i) and (ii)." Appx12. This position, however, deviated from the position adopted by the Examiner, who mapped Kiuchi's client-side proxy to the claimed DNS proxy module. *See, e.g.*, Appx1489-1493 (relying on "the client-side proxy of Kiuchi" for the claimed DNS proxy module). For example, the Examiner pointed out how "Cisco responds that the client-side proxy of Kiuchi 'will perform the recited steps for every request,'" and that "[t]he examiner agrees with the requesters [including Cisco]." Appx1493. Similarly, the Examiner explained that "Cisco responds that the client-side proxy of Kiuchi 'performs its "determining" step'" and that "[t]he examiner agrees with requesters [including Cisco]." Appx1490.[7] VirnetX did not have a fair opportunity to respond to this mapping.

On rehearing, the Board found that the Examiner did previously present this mapping. Appx124. But in doing so, the Board simply accepted Cisco's citation of instances where interaction between the client-side proxy and the C-HTTP name server was identified. *Id.* This is not the same as mapping the combination to the claimed DNS proxy module, nor is it the same thing as the rejection expressly adopted by the Examiner in the Examiner's Right of Appeal Notice ("RAN"). Appx2311-2312.

[7] The Examiner also conflictingly found that "the client-side proxy of Kiuchi" could simultaneously also be "mapped to the claimed 'client.'" Appx1492. But that would be plainly improper, and is not a position adopted by the Board. *See Mangrove*, 778 F. App'x at 906 ("The Board could not have found that the client-side proxy corresponds to the claimed 'client' and is also a part of the DNS proxy module, as the claim makes clear that these are separate components.").

The Board's reliance on this combination violates the cardinal anticipation rule that the prior-art reference must disclose "all of the limitations arranged or combined *in the same way* as recited in the claim." *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008) (emphasis added). Kiuchi's client-side proxy and C-HTTP name server stand in a client-server relationship where the former makes a request to (and receives a response from) the latter in initiating secure communications (Appx3427), which means they are not part of the same module.

The Board's decision also fails for another reason. The Board reasoned that by pointing to "the combined operation of the client-side proxy and C-HTTP name server," it could rely on how "the C-HTTP server sends a status code indicating an error, which the client-side proxy receives and then performs a DNS lookup in accordance with step (ii) in claim 1." Appx12. But the claims specifically require that "when [the / the intercepted] DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer." Appx202-203 (claims 1, 7, 13). Critically missing from Kiuchi's disclosure is any indication whatsoever as to how the DNS lookup is performed. Kiuchi never says ***what*** it sends for DNS lookup, much less that the client-side proxy ***forwards*** *the very same request that it receives*. In fact, Kiuchi is inconsistent with the claims. Kiuchi repeatedly differentiates its C-HTTP features from DNS. For example, Kiuchi explains that the C-HTTP name service is used "instead of DNS," the "DNS name service is not used for hostname resolution," and a "DNS lookup" is only performed after a permission request to the C-HTTP name server fails. Appx3426-3427; *see also* Appx3430 (explaining that C-HTTP involves a different naming scheme). Thus, in order to perform DNS lookup, the client-side proxy necessarily generates a *new* DNS request—it does not "forward[] the DNS request" it previously received, as the claims require.

On rehearing, the Board asserted that "Patent Owner does not sufficiently explain why these portions [of Kiuchi] necessarily mean that new DNS requests are generated." Appx125. This reasoning improperly shifts the burden patent owner to prove patentability. And it also ignores that the evidence only points in one direction—i.e., that new DNS requests are generated—and thus the Board's determination to the contrary is not supported by substantial evidence.[8]

For all of these reasons, the Board's rejections based on Kiuchi should be reversed. Moreover, with respect to the combination of Kiuchi and Edwards, the Board appears to have adopted these rejections solely on the basis of its finding of

[8] The Board also questioned whether VirnetX had timely made this argument. Appx125. But even Cisco acknowledged that "VirnetX's arguments on pp. 3-4 [the relevant portion of VirnetX's rehearing request] simply restate[] previous arguments that were rejected by the Examiner and the Board." Appx2313; *see also* Appx1631-1632. And to the extent there is any margin between VirnetX's arguments prior to its rehearing request below and the arguments it made in its rehearing request, those were compelled by the Board's new mapping.

anticipation based on Kiuchi. Appx36-37. As such, the rejections based on the combination of Kiuchi and Edwards should be reversed for the same reasons.[9]

B. The Board's Beser-Based Rejections Should Be Reversed

As noted above with respect to Kiuchi, the claims require that "when [the / the intercepted] DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer." Appx202-203 (claims 1, 7, 13). The combination of Beser and Kent fails to disclose this limitation.

In its request, Apple[10] argued that:

> [T]he trusted-third party network device will inherently resolve and return the IP address associated with the domain name. Also, in that embodiment, if the destination associated with the domain name is not one that will cause the trusted-third-party network device to negotiate establishment of an IP tunnel between a first and second network devices, the trusted-third-party network device will, by its nature of being a DNS server, simply return the IP address associated with the (non-secure) domain.

Appx6597. Apple does not cite a single passage in Beser (or Kent) to support this assertion. Beser in fact does not describe the trusted-third-party network device as

[9] VirnetX acknowledges that this Court affirmed similar findings by the Board in *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, No. 2020-2271, 2023 WL 2708975, at *10 (Fed. Cir. Mar. 30, 2023). VirnetX presents this argument here for preservation purposes to account for the possibility that that decision may be vacated by the Supreme Court.

[10] As discussed, the Board improperly incorporated and relied on Apple's proposed rejections in this proceeding. *Supra* at 33-39.

operating in this manner. Even if the DNS server on which the trusted-third-party network device is configured is capable of receiving and processing DNS requests, as conventional DNS servers do, this does not in any way suggest that the trusted-third-party network device would somehow have the special ability to convert tunneling requests into DNS requests "if the destination associated with the domain name is not one that will cause the trusted-third-party network device to negotiate establishment of an IP tunnel between a first and second network devices." *Id.* If such an event were to occur, the tunneling request would, at best, simply fail.

Inherent disclosure requires that "the limitation at issue necessarily must be present, or the natural result of the combination of elements explicitly disclosed by the prior art." *PAR Pharm., Inc. v. TWI Pharms., Inc.*, 773 F.3d 1186, 1196 (Fed. Cir. 2014). As this Court emphasized, "the use of inherency, a doctrine originally rooted in anticipation, must be carefully circumscribed in the context of obviousness." *Id.* at 1195. Put another way, "mere possibility [of meeting the claim limitation] is not enough." *PersonalWeb Techs., LLC v. Apple, Inc.*, 917 F.3d 1376, 1382 (Fed. Cir. 2019); *see also PAR Pharm.*, 773 F.3d at 1195 ("[t]he mere fact that a certain thing may result from a given set of circumstances" in the prior art is not enough to meet the exacting inherency standard). As a result, "[o]ccasional results" from practicing the prior art "are not inherent." *MEHL/Biophile Int'l Corp. v. Milgraum*, 192 F.3d 1362, 1365 (Fed. Cir. 1999) (holding that claims were not inherent where the prior art could be practiced in a way that did not meet the claims); *see also Transclean Corp. v. Bridgewood Servs., Inc.*, 290 F.3d 1364, 1373 (Fed. Cir. 2002) (affirming summary judgment that claims were not inherent where the prior art disclosed the missing claim element "under some circumstances," but not necessarily under all circumstances). Apple failed to meet this exacting standard.

The Examiner confusingly asserted that "[e]ven if the tunneling association were to 'simply fail' because the domain name included in the request 'does not correspond to a secure server' … that does not mean the DNS function of the trusted-third-party network device 30 would not return an IP address." Appx1485. But that is pure speculation unsupported by any evidence whatsoever, and again fails to meet the exacting standard required for inherency.

The inherency finding for Beser is particularly unsupported given the way Beser functions. In Beser, when an originating device wants to communicate with a terminating device, it sends a tunnel initiation request to the first network device. Appx4722 (7:65-66). This request "includes a unique identifier for the terminating end of the tunneling association." Appx4722 (8:1-3). The first network device then informs the trusted-third-party network device of the request to initiate a tunnel. Appx4722 (8:48-49); Appx4724 (11:9-12); *see also* Appx4707 (Fig. 6). Thus, the trusted-third-party network device is only reached in the case of a specialized tunneling request, and there is no evidence to suggest that the trusted-third-party network device would somehow have the special ability to convert failed tunneling requests into different DNS requests that cause the return of an IP address.

The Board did not even substantively address this issue. Instead, the Board proclaimed (without explanation) that "Patent Owner does not meaningfully address the Examiner's rationale." Appx44. That was plainly incorrect—VirnetX addressed this issue at length. Appx1617-1618. For all of these reasons, the Board's rejections based on Beser and Kent should be reversed.[11]

C. The Board's Aventail/AutoSOCKS-Based Rejections Should Be Reversed

The Board's rejections based on Aventail and AutoSOCKS[12] are based on a faulty construction of the claimed "secure channel," Appx203 (claim 13), which the claims and specification make clear is a direct channel that is secure. In particular, the '151 patent describes *direct* communications between a client device and a target device. For instance, in one embodiment, the '151 patent describes the communication between an originating TARP terminal and a destination TARP terminal as direct. Appx183 (7:34-43); Appx146 (Fig. 2); *see also* Appx195 (31:30-

[11] A substantively identical issue is also present in Appeal No. 22-1997 (i.e., the appeal involving reexamination no. 95/001,697, which is the proceeding where Apple made this argument in the first place that was later merged into this proceeding).

[12] As noted above, the Board indicated that "[t]here is no dispute that the disclosure of AutoSOCKS is 'substantially similar' to Aventail Connect v3.01." Appx25. Given that both references were addressed in an identical manner below, VirnetX addresses them collectively here as well.

39) (describing a variation of the TARP embodiments as including a direct communication link); Appx197 (35:64-67) (describing the embodiment of Figure 24 in which a first computer and second computer are connected directly). The '151 patent similarly describes direct communications in later embodiments as well. Appx198 (37:66-38:2, 38:59-62) (describing a virtual private network as being direct between a user's computer and target); Appx199 (39:66-40:3, 40:58-62) (describing a load balancing example in which a virtual private network is direct between a first host and a second host).

In each of these embodiments, the '151 patent specification discloses that the communication traverses a network (or networks) through which it is simply passed or routed via various network devices such as Internet Service Providers, firewalls, and routers. Indeed, in related litigation involving the same patent, Cisco and its co-defendants recognized that this type of network traversal is a "direct" communication. *See, e.g.*, *VirnetX Inc. v. Cisco Sys., Inc.*, No. 6:10-cv-00417, Docket No. 289 (E.D. Tex. May 18, 2012) (Jan. 5, 2012 *Markman* Hr'g Tr.) at 2:16-21, 4:17-5:12; *see also id.* at 44:13-45:12 (explaining that the claims should be limited to "direct" communication because the specification teaches direct communication between the client and target). In view of this, this Court has previously construed the related terms "secure communication link" and "virtual private network" to include "direct" communication. *Cisco*, 767 F.3d at 1317 n.1,

1319. Accordingly, "secure channel" should be construed as a "direct channel that is secure."

Aventail does not disclose automatically initiating such a *direct* channel between an application (alleged client) and a remote host (alleged secure server). The Examiner-described "path to the remote host" from the client is not a direct channel because as the Office concedes, the alleged communications channel in Aventail passes through the SOCKS server. Appx1473 (alleging that Aventail "teaches that the channel is encrypted at least from the application to the Aventail ExtraNet Server (a proxy server) along the path to the remote host"). But this path from the client to the remote host is relayed through the SOCKS server and is not a direct connection between the client device and the remote host. This is because the connection is terminated at the SOCKS server, which may create a new connection to the remote host. Such an operation is evident from Aventail's disclosure that Aventail Connect routes appropriate network traffic from the client application to the SOCKS server. Appx4558. That is, the traffic is addressed to the SOCKS server and "[t]he SOCKS server *then* sends the traffic to the Internet or the external network" depending on the rules defined by an administrator for that incoming or outgoing traffic. Appx4558 (emphasis added); Appx4564 (step 3, "Aventail Connect encrypts the data on its way to the server").) This operation is different from a regular firewall or an edge router, where the traffic is destined for the target device and not for the firewall or the edge router. Indeed, Aventail distinguishes the SOCKS server from a regular firewall stating that "SOCKS is more than a standard security firewall." Appx4557-4558.

The Board failed to give proper consideration to VirnetX's argument that "secure channel" should be construed as a "direct channel that is secure," and the resulting consequences of that construction. Instead, the Board only observed that in a different case, where this argument was not at issue, "the Federal Circuit did not indicate that a 'direct' channel was required in the '151 Patent." Appx22. The Board then went on to conclude that "[e]ven if the claims of the '151 Patent do require a 'direct' channel, Patent Owner has not sufficiently explained how such a requirement in the claims necessarily distinguishes Aventail Connect v3.01." *Id.* This finding, however, defies this Court's prior characterization of Aventail, where Aventail was described as including a system involving an indirect communication "in which a client computer communicates with an intermediate server via a singular, point-to-point connection" and the "intermediate server then relays the data to a target computer." *Mangrove*, 778 F. App'x at 910.

For all of these reasons, the Board's rejections based on Aventail and AutoSOCKS should be reversed.

D. The Board's Aziz-Based Rejections Should Be Reversed

The claims require "determining whether [the intercepted / a] DNS request corresponds to a secure server." Appx202-203 (claims 1, 7, 13). The Board's determination that the combination of Aziz and Edwards discloses this limitation lacks substantial evidence, and should be reversed.

The Board relied on Aziz for this limitation. In particular, the Board asserted that "Aziz discloses the data field of the SX record contains 'the identifier (e.g., name or address) of a 'secure exchanger' associated with the owner of the record,' where the 'secure exchanger is a machine that handles secure communications for itself or for another machine (e.g., performs encryption or decryption).'" Appx32 (citing Appx3490 (6:27-32); Appx3492 (10:42-52)). According to the Board, this teaching was sufficient to meet the claim limitation because "the '151 Patent discloses that determining whether a secure site has been requested is through 'domain name extension, or by reference to an internal table of such sites.'" Appx32 (citing Appx198 (37:62-66)).

Contrary to the Board's finding, Aziz does not disclose any determination about whether the DNS request corresponds to a secure server. Aziz explains that "outside NS" 120 may receive a query for a host address located within domain 100 and may check its database for an SX record and the requested host name. Appx3492 (9:49-53). But Aziz does not disclose that the existence of an SX record dictates that the matching host name corresponds to a secure server. Appx5872 (¶ 166). For example, *Aziz* explains that SX records are just another type of resource record. Appx3490 (6:25-28). And NS 120 behaves like any other name server-when it receives a request; it checks to see what types of records it has for the name in the request, such as address records ("A records") or SX records. Appx5871-5872 (¶ 165). If it has an SX record, it simply adds it to the response it will send to the client, much like it does with A records:

> At step 310, outside NS 120 *checks if its zone database has an SX record* with an owner name that matches the requested host name. If the database does not have such a record, execution jumps to step 320. If the database does, at step 315, outside NS 120 adds the SX record identifying the secure exchanger for the requested host to the response.

Appx3492 (9:49-55). Similarly, in Figure 3, Aziz shows that NS 120 merely checks for the existence of an SX record in step 310, but does not determine whether a website is secure.

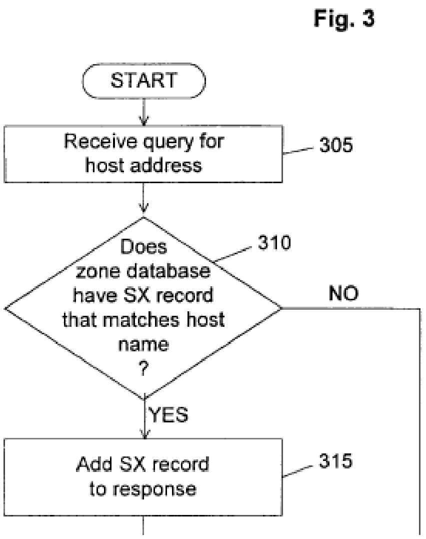


Fig. 3
START
Receive query for host address
305
Does zone database have SX record that matches host name ?
310
NO
YES
Add SX record to response
315

Appx3480 (Fig. 3).

Nor is the Board's comparison of Aziz to the '151 patent appropriate. Appx32 (citing Appx198 (37:62-66)). The table described in the '151 patent and identified by the Board is expressly a "table of such sites," i.e., a table of secure sites. Appx198 (37:62-66). On the other hand, Aziz does not disclose that the existence of an SX record dictates that the matching host name corresponds to a secure server. Appx5872 (¶ 166). Just because an SX record may be used for secure communications does not mean that all host names with SX records correspond to secure servers and host names without SX records do not correspond to secure servers. Appx5872 (¶ 166). Instead, Aziz explains that the SX record is merely a resource record that stores the name and address of a secure exchanger, such as a firewall. Appx3490 (6:25-28); Appx5872 (¶ 167). Aziz does not disclose that the SX record indicates whether a corresponding server is secure. Appx5872 (¶ 167).

For all of these reasons, the Board's rejections based on Aziz should be reversed.

CONCLUSION

The Board's final written decision should be reversed or, at a minimum, vacated and remanded.

February 5, 2024

Respectfully submitted,

/s/ Naveen Modi
Naveen Modi
Stephen B. Kinnaird
Joseph E. Palys
Igor V. Timofeyev
Daniel Zeilberger
PAUL HASTINGS LLP
2050 M Street, N.W.
Washington, D.C. 20036
(202) 551-1700
(202) 551-1705 (fax)
naveenmodi@paulhastings.com

Counsel for Appellant VirnetX Inc.

ADDENDUM

ADDENDUM TABLE OF CONTENTS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/001,714	08/16/2011	7,490,151 B2	43614.99	3428

137313 7590 06/23/2020
PAUL HASTINGS LLP
875 15th Street, NW
Washington, DC 20005

EXAMINER
YIGDALL, MICHAEL J

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
06/23/2020	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.
Requester and Cross-Appellant

v.

Patent of VIRNETX INC.
Patent Owner and Appellant

Appeal 2020-000639
Reexamination Control 95/001,714
Patent 7,490,151 B2
Technology Center 3900

Before JEFFREY B. ROBERTSON, DENISE M. POTHIER, and JEREMY J. CURCURI, *Administrative Patent Judges.*

ROBERTSON, *Administrative Patent Judge.*

DECISION ON APPEAL

I. STATEMENT OF THE CASE[1]

VirnetX Inc. ("Patent Owner") appeals under 35 U.S.C. §§ 134(b) and 315(a) (Pre-AIA) from the Examiner's decision to reject claims 1–16.[2] Third-Party Requester Cisco Systems Inc. (hereinafter "Cisco") urges that

[1] The instant reexamination proceeding was merged with Reexamination Control No. 95/001,697 filed by Apple, Inc. in a DECISION *SUA SPONTE* MERGING REEXAMINATION PROCEEDINGS entered March 15, 2012. In a "DECISION GRANTING RENEWED PETITION TO SEVER MERGER AND RENEWED PETITION TO TERMINATE REEXAMINATION PROCEEDING" entered October 16, 2019 ("Decision Severing Merger"), the merger was severed. The Decision Severing Merger concluded only that any rejection applied against claims 1–6 and 13–16 will not be further maintained by the Office "in the '1697 reexamination proceeding." Decision Severing Merger 15. This decision, however, considers rejections maintained by the Office against claims 1–6 and 13–16 for rejections as part of Reexamination Control No. 95/001,714 even though certain rejections originated from Reexamination Control No. 95/001,697. *See* DECISION ON JULY 9, 2013 PATENT OWNER PETITION FOR RELIEF FROM MAY 24, 2013 DECISION, July 7, 2014, 13–14 (explaining that Cisco is entitled to file a respondent brief that can address any of the Apple issues raised in patent owner's appellant brief because such are directed to the examiner's adopted rejections originally proposed by Apple addressed in patent owner's appellant brief); DECISION DENYING PATENT OWNER'S APRIL 2014 PETITION FOR REVIEW OF DECISION MAINTAINING MERGER OF REEXAMINATION PROCEEDINGS, September 21, 2015 13–15 (explaining the limits on appeal in merged proceedings); DECISION ON PETITIONS July 17, 2017 12 (explaining that the respondent's brief may include any arguments previously made of record that support the examiner's findings with respect to any claim addressed in the opposing party's appellant brief).

[2] *See* Patent Owner's Appeal Brief filed June 6, 2013 (hereinafter "PO Appeal Br.") vi; Examiner's Answer (mailed March 30, 2017) (hereinafter "Ans."); Right of Appeal Notice (mailed February 24, 2016) (hereinafter "RAN"); Patent Owner's Rebuttal Brief filed May 1, 2017 (hereinafter "PO Reb. Br.)".

the Examiner's decision must be affirmed.[3] Cisco cross-appeals under 35 U.S.C. §§ 134(c) and 315(b) from the Examiner's decision to withdraw several rejections of claims 1–16 on various grounds.[4] Patent Owner urges that the Examiner's decision must be affirmed.[5] We have jurisdiction under 35 U.S.C. §§ 134(b)–(c) and 315(a)–(b) (Pre-AIA).

We affirm-in-part the Examiner's decision to reject certain claims and to withdraw certain rejections. We reverse the Examiner's decision not to reject certain claims. By operation of rule, our reversal of the Examiner's decision not to reject certain claims is designated a new ground of rejection. 37 C.F.R. § 41.77(b).

II. INTRODUCTION

A. Background and Summary

United States Patent 7,490,151 B2 (hereinafter the "'151 Patent"), which is the subject of the current *inter partes* reexamination, issued to Munger et al. on February 10, 2009. *Inter partes* Reexamination was requested by Cisco ("REQUEST FOR INTER PARTES REEXAMINATION" filed on August 16, 2011, "Request"). Both Patent Owner and Cisco identify numerous related appeals and proceedings. *See* PO Appeal Br. iii–vi; Req. Appeal Br. 1–2. In particular, the Federal Circuit, in *VirnetX v. Mangrove Partners Master Fund, Ltd.*, 778 F. App'x

[3] *See* Cisco's Respondent Brief (filed August 18, 2016) (hereinafter "Cisco Resp't Br.").

[4] *See* Cisco's Appeal Brief filed June 3, 2016 (hereinafter "Cisco Appeal Br.") 3–4; Cisco's Rebuttal Brief filed May 1, 2017 (hereinafter "Cisco Reb. Br.").

[5] *See* Patent Owner's Respondent Brief filed February 27, 2017 (hereinafter "PO Resp't Br.").

897 (Fed. Cir. 2019), vacated and remanded a PTAB Final Written Decision in IPR2015-01047, an *inter partes* review of the '151 Patent.

The claims of the '151 Patent relate to secure and non-secure communications in response to a domain-name server look-up function. *See* '151 Patent, col. 36, l. 55 – col. 39, l. 46.

Claim 1, which is illustrative of the appealed subject matter, reads as follows:

> 1. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:
>
> (i) determining whether the intercepted DNS request corresponds to a secure server;
>
> (ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and
>
> (iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

(PO Appeal Br. i, Claims App.)

B. *Adopted Rejections*

Patent Owner contests the Examiner's decision to reject the claims as follows (PO Appeal Br. 3; *see* RAN 7–41; Ans. 2–170):

Claim(s) Rejected	35 U.S.C. §	Reference(s)
1, 2, 5, 7, 8, 11, 13, 14	102(b)	Aventail Connect v3.01[6]
1, 2, 5, 7, 8, 11, 13, 14	102(b)	AutoSOCKS[7]
1, 2, 5, 7, 8, 11, 13, 14	103(a)	Beser,[8] Kent[9]
1–4, 7–10, 13–16	102(b)	Kiuchi[10]
5, 11	103(a)	Kiuchi, Martin[11]
1, 7, 13	103(a)	Aziz,[12] Edwards[13]
5, 11	103(a)	Aziz, Edwards, Martin
1–4, 6–10, 12–16	103(a)	Kiuchi, Edwards
5, 11	103(a)	Kiuchi, Edwards, Martin

[6] *Aventail Connect v3.01/v2.51 Administrator's Guide,* 1999 ("Aventail Connect v3.01").

[7] *Aventail AutoSOCKS v2.1 Administration & User's Guide,* 1997 ("AutoSOCKS").

[8] Beser et al., U.S. Patent No. 6,496,867, issued December 17, 2002 ("Beser").

[9] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," Network Working Group RFC 2401, November 1998 ("Kent").

[10] Takahiro Kiuchi and Shigekoto Kaihara, C-HTTP – *The Development of a Secure, Closed HTTP-Based Network on the Internet*, PROCEEDINGS OF THE SYMPOSIUM ON NETWORK AND DISTRIBUTED SYSTEM SECURITY, IEEE 64–75 (1996) ("Kiuchi").

[11] David M. Martin Jr., *A Framework for Local Anonymity in the Internet*, Technical Report. Boston University, Boston, MA. (February 21, 1998) ("Martin").

[12] Ashar Aziz et al., U.S. Patent No. 6,119,234, issued September 12, 2000 ("Aziz").

[13] Nigel Edwards and Owen Rees, *High security Web servers and gateways*, "Computer Networks and ISDN Systems" 29 (Sept. 1997), pp. 927–938 ("Edwards").

C. Non-Adopted Rejections

Cisco contests the Examiner's decision to withdraw rejections of the claims as follows (Cisco Appeal Br. 2):

Claim(s) not Rejected	35 U.S.C. §	Reference(s)
6, 12	102(b)	Kiuchi
1–4, 6–10, 12–16	102(e)	Wesinger[14]
5, 11	103(a)	Wesinger, Martin
1, 7, 13	102(e)	Blum[15]
2–4, 6, 8–10, 12, 14–16	103(a)	Aziz, Edwards
1–4, 6–10, 12–16	103(a)	Wesinger, Edwards
5, 11	103(a)	Wesinger, Edwards, Martin

III. PATENT OWNER'S APPEAL

A. Anticipation – Kiuchi (Issue 7)[16]

1. The Examiner's Findings

The Examiner rejected claims 1–4, 7–10, and 13–16 as anticipated by Kiuchi. *See* Ans. 62–86. The Examiner found Kiuchi discloses a client (an end-user) selects and requests connection to a resource or server, which request is intercepted by the client-side proxy. *Id.* at 64–66, citing Kiuchi 64–65. The Examiner found Kiuchi discloses a client-side proxy stored on a

[14] Wesinger, Jr. et al., U.S. Patent No. 5,898,830, issued April 27, 1999 ("Wesinger").
[15] Blum et al., U.S. Patent No. 6,182,141 B1, issued January 30, 2001 ("Blum").
[16] The designation "Issue 7" is based on the labeling of rejections by the Patent Owner, Cisco, and the Examiner. We refer to these designations for convenience of the reader, although we address the "Issues" in a different order.

firewall. Ans. 65, citing Kiuchi 68. The Examiner found the client-side proxy disclosed in Kiuchi responds to a domain name inquiry by attempting to resolve the domain name to an IP address in two different ways; by (1) first requesting the IP address from a closed Hypertext Transfer Protocol-based network (C-HTTP), and if that fails, (2) requesting the IP address from an ordinary DNS. *Id.* at 65–66, citing Kiuchi 65.

As to the step of determining whether the intercepted DNS request corresponds to a secure server (the "determining step") recited in claim 1, the Examiner found Kiuchi discloses the C-HTTP-based secure server examines whether a requested server-side proxy is registered in the closed C-HTTP-based network, and when it is not, the client-side proxy receives an error status and then performs DNS lookup. *Id.* at 66–69, citing Kiuchi 64, 65, 68.

The Examiner found also Kiuchi discloses "automatically initiating an encrypted channel between the client and the secure server" as recited in claim 1. *Id.* at 71–72, citing Kiuchi 65–66. That is, the Examiner found Kiuchi discloses when the C-HTTP name server confirms the server-side proxy is an appropriate closed network member, the client-side proxy sends a request for connection, which is encrypted using the server-side proxy's public key, and that "following this request for connection, the client-side proxy (the 'client') and the server-side proxy (the 'secure server') exchange messages . . . Since the connection between the client-side proxy and the server-side proxy is established without any user involvement, the connection is initiated 'automatically.'" *Id.* at 72, citing Kiuchi 66. The Examiner found Kiuchi discloses the recited steps are performed for each request. *Id.* at 67.

2. *Patent Owner's Contentions*

Patent Owner contends Kiuchi fails to disclose a DNS proxy module, because Kiuchi discloses expressly that the closed HTTP-based network (C-HTTP) for communication does not involve DNS. PO Appeal Br. 35–37, citing Kiuchi 64, 68; Declaration of Angelo D. Keromytis dated May 8, 2012 ("Keromytis Decl."), ¶ 84. Patent Owner also contends Kiuchi fails to disclose the step of "determining whether the intercepted DNS request corresponds to a secure server" as recited in claim 1, because the rejection maps the "client-side proxy" to the DNS proxy module, but then maps the recited "determining" step to the "C-HTTP name server" disclosed in Kiuchi. PO Appeal Br. 37–38. Patent Owner argues Kiuchi does not disclose "automatically initiating an encrypted channel between the client and the secure server" as recited in claim 1. PO Appeal Br. 38–41. Patent Owner argues Kiuchi fails to disclose a DNS proxy module that performs the steps recited in claim 1 "for each intercepted DNS Request." PO Appeal Br. 41–42 (emphasis omitted).

3. *Cisco's Contentions*

Cisco responds that Patent Owner has provided no particular definition, claim interpretation, or substantive analysis that would differentiate the recited "DNS proxy module" and "DNS request" from Kiuchi's disclosure of a "domain name service" that "performs DNS lookup, behaving like an ordinary HTTP/1.0 proxy." Cisco Resp't Br. 14–16, citing Kiuchi 65. Cisco contends the client-side proxy performs the "determining step" by sending a request to the C-HTTP name server and evaluating the response received, such that the client-side proxy behaves differently depending on the received response of either receiving a secure network address, or an error message. *Id.* at 16–17. Cisco contends also that in the Examiner's analysis, the client-side proxy corresponds to the recited "client" in claim 1. *Id.* at 17–18. Cisco argues also that Kiuchi discloses that when a series of different target destinations is requested, Kiuchi discloses the client-side proxy will perform the recited steps of every request, initiating a new encrypted channel for each request. *Id.* at 18–19.

4. Analysis

We limit our discussion to claims 1 and 13–15, which is sufficient to resolve the issues associated with this rejection. *See* 37 C.F.R. § 41.67(c)(vii).

a) "Client"

We begin by addressing the Examiner's mapping of Kiuchi to the "client" recited in claim 1. In particular, as discussed above, we observe that at different points, the Examiner, in adopting Cisco's rejections, has mapped both the "user agent" as well as the "client-side proxy" disclosed in Kiuchi to the "client" recited in claim 1. *See* RAN 28–29. The Examiner's position is that regardless of whether the "user agent" or the "client-side proxy" disclosed in Kiuchi corresponds to the "client" in claim 1, claim 1 is anticipated by Kiuchi. *Id.* at 29.

We observe that the Federal Circuit, in summarizing Kiuchi, has previously equated the "user agent" disclosed in Kiuchi with a client. *VirnetX*, 778 F App'x at 905 ("[Kiuchi's] system consists of five relevant elements: a user agent (also referred to as a client), a client-side proxy, a C-HTTP name server, a server-side proxy, and an origin server."). In this regard, the Federal Circuit stated in a similar situation where Kiuchi's user agent and client-side proxy were both relied upon to correspond to the recited "client" that "[t]here is no question that these are different components in Kiuchi's system," and that a reliance on different components in Kiuchi as a disclosure of the recited "client" is not supported by substantial evidence. *Id.* at 778 F App'x at 907–908.

We are of the view that the "user agent" of Kiuchi corresponds to the "client" recited in claim 1. Kiuchi discloses a user agent requests an "HTML document" or "HTTP/1.0 request" that is encrypted and wrapped in a C-HTTP request by the client-side proxy. Kiuchi 65.

b) "Domain Name Server (DNS)"

We next address Patent Owner's argument that Kiuchi's C-HTTP techniques do not involve DNS. Cisco identifies several proceedings where the Patent Office has rejected Patent Owner's argument as insufficient to distinguish the services provided by Kiuchi's C-HTTP name server from DNS requests sent by a client. Cisco Resp't Br. 15. Although Kiuchi discloses a C-HTTP-based name service is used "instead of DNS" (Kiuchi, Abstr.), Patent Owner's arguments fail to identify a substantive difference between the DNS request recited in claim 1 and the user agent's request in Kiuchi, where the client-side proxy may perform a DNS lookup. *See* Kiuchi 65–66, Fig. c.[17] Thus, we are persuaded by the evidence that the resource name intercepted by the client-side proxy in Kiuchi is encompassed by the DNS request intercepted by the DNS proxy server recited in claim 1.

[17] We observe also that Patent Owner made similar arguments in IPR2015-01047, which the Board found to be unpersuasive. *Mangrove Partners Master Fund, LTD, Apple Inc., and Black Swamp IP, LLC v. VirnetX, Inc.*, IPR2015-01047, Paper 80 at 6–8 (PTAB Sept. 9, 2016) (Final Written Decision).

c) Each Intercepted DNS Request

We are not persuaded by Patent Owner's argument that Kiuchi fails to disclose performing the recited steps for each intercepted DNS request as recited in claim 1. In particular, as Cisco points out (Req. Resp't Br. 18–19), Kiuchi discloses a TCP connection is closed after each transaction (request and request pair). Kiuchi 65, 67. Thus, Kiuchi discloses performing the steps recited for each DNS request.

d) Steps (i) and (ii)

We are not persuaded that Kiuchi fails to disclose the recited "determining step" (i) in claim 1. At the outset, we clarify that the combination of the "client-side proxy" and the "C-HTTP name server" disclosed in Kiuchi corresponds to the DNS proxy module that performs steps (i) and (ii) recited in claim 1. That is, in Kiuchi, the client-side proxy asks the C-HTTP name server whether it can communicate with the host specified in a given URL, and if communication is not permitted, the C-HTTP server sends a status code indicating an error, which the client-side proxy receives and then performs a DNS lookup in accordance with step (ii) in claim 1. Kiuchi 65. Thus, the combined operation of the client-side proxy and C-HTTP name server in Kiuchi performs steps (i) and (ii) in claim 1. *See* Ans. 67–70, Chart E-1.1 (discussing how "examining 'whether the requested server-side proxy is registered in the closed network' shows determining whether the intercepted DNS request corresponds to a secure server a recited in the claim" and "[p]erforming 'DNS lookup' on requests that are not part of the closed network (i.e., do not correspond to a 'secure server' in the closed network) shows forwarding the DNS request to a DNS function as recited by the claim.").

e) Step (iii) "automatically initiating an encrypted channel between the client and the secure server"

We are not persuaded by Patent Owner's arguments that Kiuchi is silent as to whether the C-HTTP connections are established automatically.[18] PO Appeal Br. 38–39; Keromytis Decl. ¶¶ 88–89. Rather, we agree with the Examiner that Kiuchi, by disclosing the client-side proxy sends a request for connection to the server-side proxy and the server-side proxy sends a symmetric data exchange key for response encryption to the client-side proxy, after which connection is established, discloses "automatically initiating" as recited in the claims. RAN 28–29, citing Kiuchi 65–66. Thus, Patent Owner's assertions with respect to possible user agent involvement in establishing connections are not supported by the record.

f) Claims 1–4 and 7–10

As to the recitation in claim 1 that the encrypted channel is initiated "between the client and the secure server," the Examiner's view is that if the "user agent" in Kiuchi corresponds to the "client" recited in claim 1, the channel between the "user agent" and the "server-side proxy" is an encrypted channel, because requests from the user agent are forwarded in encrypted form from the client-side proxy to the server-side proxy and

[18] In view of the discussion of the Federal Circuit's Decision in *VirnetX v. Mangrove Partners Master Fund, Ltd.*, 778 F. App'x 897 above, we find it unnecessary to discuss Cisco's contention that Patent Owner is prohibited from arguing Kiuchi's disclosure of an encrypted link between the client-side proxy and the server-side proxy is distinguishable from the recitation in claim 1, "automatically initiating an encrypted channel between the client and the secure server," under 37 C.F.R. § 42.73, because in a final written decision in IPR2014-00482, the Board found the same argument unpersuasive. Cisco Resp't Br. 17.

responses to the user agent are forwarded from the server-side proxy to the client-side proxy "encrypted in C-HTTP format." RAN 29. In view of the discussion above, however, we are persuaded by Patent Owner's argument that Kiuchi discloses an encrypted link only between the client-side proxy and the server-side proxy. Appeal Br. 39. Kiuchi discloses only that communication between the client-side proxy and server-side proxy is encrypted while communications between a user agent and the client-side proxy or an origin server and server side proxy are performed "using current HTTP/1.0." Kiuchi, Abstract.

Accordingly, we reverse the Examiner's rejection of claims 1–4 as anticipated by Kiuchi. Because independent claim 7 contains similar language, we reverse the Examiner's rejection of claims 7–10 as anticipated by Kiuchi for similar reasons.

g) Claim 13

Independent claim 13 recites "when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel between the client and the secure server." Thus, claim 13 does not require "encryption" between the client and the secure server. Kiuchi discloses "each member" of a closed group of institutions on the Internet "is protected by its own firewall." Kiuchi, Abstract, 64 § 2.1, 67 § 4.2. Therefore, we find Kiuchi discloses a secure channel as required by claim 13.

Accordingly, we affirm the Examiner's rejection of claim 13.

h) Claim 14

Claim 14 depends from claim 13 and recites, *inter alia*, "determining whether the client is authorized to access the secure server."

Patent Owner contends Kiuchi does not disclose determining whether the client is authorized to access a secure server because the client-side proxy is not synonymous with client, and while firewalls or proxies between two institutions may be authorized to communicate, individual clients and users within each institution may or may not be authorized to do so. PO Appeal Br. 43–44.

We are not persuaded by Patent Owner's argument. Rather, we agree with the Examiner's analysis, that the client-side proxy in Kiuchi, acting as a proxy for the client, determines whether the connection between the client and the secure server is permitted or not permitted, which is encompassed within the scope of "authorized" as recited in claim 14. RAN 32, citing Kiuchi 65; *see also* Kiuchi 65–66, disclosing the C-HTTP server determines whether the client-side proxy is an appropriate member of the closed network.

i) Claim 15

Claim 15 depends from claim 14, and recites "wherein step (iii) further comprises the step of: (c) when the client is not authorized to access the secure server, returning a host unknown error message to the client."

The Examiner found Kiuchi discloses an error message when the connection between the client-side proxy and server-side proxy is not permitted. Ans. 74–76, 85, citing Kiuchi 65, 68, and RFC 1035 at 27.[19]

Patent Owner contends the C-HTTP name server sends a status code that indicates an error, rather than the client-side proxy identified as the

[19] Mockapetris, P. RFC 1035, "Domain Names-Implementation and Specification," November 1987. Cisco Resp't Br., Ex. D8.

alleged DNS proxy module. PO Appeal Br. 45. Patent Owner argues also the error message is not returned to the client, but rather the client-side proxy. PO Appeal Br. 46. Patent Owner argues Cisco and the Office improperly rely on another reference, RFC 1035, in order support the position that DNS query formats and response codes include "host not found," and thus, the rejection "improperly relies on more than one reference for the claimed features" of claim 15. PO Appeal Br. 47.

We are persuaded by Patent Owner's arguments. That is, Kiuchi discloses only that the client-side proxy receives an error status when the C-HTTP name server determined the connection is not permitted, prompting it to perform a DNS lookup. Kiuchi 68. There is no indication that such an error message would be returned from the client-side proxy to the client or user agent discussed in Kiuchi. Thus, even if we were to agree that the Examiner's reliance on RFC 1035 is appropriate, such is insufficient to support the position that a "host unknown" error message would be returned to the client in the context of Kiuchi's system.

Accordingly, we reverse the Examiner's rejection of claim 15. Because claim 16 depends from claim 15, we reverse the Examiner's rejection of claim 16 as well.

In sum, we affirm the Examiner's decision to reject claims 13 and 14 as anticipated by Kiuchi. However, we reverse the Examiner's decision to reject claims 1–4, 7–10, 15, and 16 as anticipated by Kiuchi.

B. Obviousness – Claims 5 and 11 over Kiuchi and Martin (Issue 8)

Claims 5 and 11 depend from claims 1 and 7, respectively, and recite the step of initiating encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and secure server. Martin is cited for disclosing an IP address hopping scheme that can co-exist with the system of Kiuchi. Ans. 90. Martin fails to cure the deficiencies identified in Kiuchi for claims 1 and 7. As a result, we reverse the Examiner's decision to reject claims 5 and 11 for similar reasons as discussed above with respect to claims 1 and 7.

C. Anticipation – Aventail Connect v3.01 (Issue 1)

1. The Examiner's Findings

The Examiner found, *inter alia*, Aventail Connect v3.01 discloses a proxy module that intercepts network traffic to and from a client application including DNS requests from a client application, where Aventail Connect encrypts the data on its way to the server on behalf of the application. Ans. 3–4, citing Aventail Connect v3.01 7, 12. The Examiner found Aventail Connect v3.01 discloses the channel is encrypted "at least from the application to the Aventail ExtraNet Server (a proxy server) along the path to the remote host." RAN 10. The Examiner found: "[s]pecifically, the channel between the application and the remote host is an 'encrypted channel' because the data on the channel is encrypted 'on its way' to the proxy server and when 'being returned' to the application." *Id.*

2. Patent Owner's Contentions

Patent Owner contends that Aventail Connect v3.01 has not been shown to be prior art, because there is insufficient evidence of public accessibility. PO Appeal Br. 4–7. In particular, Patent Owner argues the Declarations relied upon by the Examiner as evidence of public accessibility are uncorroborated. *Id.* at 4, citing, e.g., Declaration of James Chester.[20] Patent Owner argues the Examiner's reliance on the copyright date of 1999 in Aventail Connect v3.01 is insufficient to establish the reference as a printed publication. *Id.* at 7.

Substantively, Patent Owner contends Aventail Connect v3.01 does not disclose or suggest encrypted connections between the client and the secure server, and therefore does not disclose "automatically initiating an encrypted channel between the client and the secure server" as recited in claims 1 and 7. PO Appeal Br. 12–16. Patent Owner argues also Aventail Connect v3.01 does not disclose automatically initiating a direct channel that is encrypted between an application and a remote host, but rather the path is relayed through a server (SOCKS server). *Id.* at 16–17. Patent Owner relies on similar arguments with respect to independent claim 13. *Id.* at 17.

3. Cisco's Contentions

Cisco contends the status of Aventail Connect v3.01 as publically accessible is confirmed at least by the Chester Declaration in combination with the copyright date on Aventail Connect v3.01. Cisco Resp't Br. 5–6.

Cisco contends Patent Owner has not established that the claims require encryption extending to the computer/remote host. Cisco Resp't Br. 8–9.

[20] *See* Cisco Resp't Br., Evidence App., 4 citing Apple Ex. E3 Declaration of James Chester filed July 25, 2011 ("Chester Declaration").

4. Analysis

We limit our discussion to claims 1 and 13–14, which is sufficient to resolve the issues associated with this rejection. *See* 37 C.F.R. § 41.67(c)(vii).

a) Status of Aventail Connect v3.01 as Prior Art

We are not persuaded by Patent Owner's contentions that Aventail Connect v3.01 does not qualify as a printed publication due to lack of public accessibility. Aventail Connect v3.01 includes disclosures that it is a publically distributed document by including a 1996–1999 copyright notice by Aventail Corporation, a website http://www.aventail.com, and the statement "[p]rinted in the United States of America." Aventail Connect v3.01, i. Aventail Connect v3.01 provides contact information for "Aventail Technical Support." *Id.* at 5. Aventail Connect v3.01 lists Aventail protected trademarks and copyrights, the Aventail mailing and email addresses, further evidencing that Aventail Connect v3.01 is the kind of document expected to be widely disseminated. *Id.* at i.

We determine the Chester Declaration provides sufficient evidence that Aventail Connect v3.01 was accessible to the public and thus constitutes a printed publication. In particular, Mr. Chester declared that as an employee of Internal Business Machines Corporation (IBM) from 1992–2002, he evaluated network security products from vendors, and that he received a number of Aventail products. Chester Decl. ¶¶ 5–6. Mr. Chester similarly "recall[ed] that Aventail announced its AEC v3.0 product in the fall of 1998, and began distributing this product no later than mid-January 1999." *Id.* at ¶ 15. Mr. Chester declared that the AECv3.0 product included version 3.01/2.51 of the Aventail Connect software, which he received "no later than July 1998." Chester Decl. ¶¶ 16–17, Ex. C. He also testified IBM "deployed VPN solutions based on this product to more than 20,000 IBM employees domestically by March 1998 and more that 65,000 IBM employees worldwide by July 1998." *Id.* ¶ 19.

Even if corroboration is required to show public availability, "[c]orroboration does not require that every detail of the testimony be independently and conclusively supported by explicit disclosures in the pre-critical date documents or physical exhibits." *Ohio Willow Wood Co. v. Alps S., LLC*, 735 F.3d 1333, 1348 (Fed. Cir. 2013). *Willow Wood* stated a "rule of reason" test in which "the totality of the evidence . . . including circumstantial evidence" is assessed "in order to ascertain whether the testimonial assertions are credible." *Id.* "A given reference is 'publicly accessible' upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it." *SRI Int'l, Inc. v. Internet Sec. Sys., Inc.*, 511 F.3d 1186, 1194 (Fed. Cir. 2008) (quoting *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006)).

The evidence as outlined above supports a finding that persons interested and ordinarily skilled in the subject matter of Aventail Connect v3.01 could have located them. The evidence shows that Aventail Corp. announced release of the products and they were disseminated with the manuals. Mr. Chester declared generally that "the Aventail products were distributed with installation discs and printed manuals" and of the specific availability of Aventail Connect v3.01. Chester Decl. ¶¶ 12, 15–17. Therefore, the evidence as a whole supports the Examiner's finding that

Aventail Connect v3.01was publicly accessible. Accordingly, we agree with the Examiner and Cisco that Aventail Connect v3.01 qualifies as a prior art printed publication under 35 U.S.C. § 102(b).

b) Claims 1 and 7

Regarding claims 1 and 7, we agree with Patent Owner that Aventail Connect v3.01 does not disclose "automatically initiating an encrypted channel between the client and the secure server" as recited in claims 1 and 7. Similar to the discussion above with respect to Kiuchi, encryption and decryption occurs at Aventail Connect (Aventail Connect v3.01, 1, 12), which corresponds to the DNS proxy module recited in claims 1 and 7. As such, Aventail Connect v3.01 does not disclose initiating an encrypted channel "between the client and the secure server" as recited in claims 1 and 7. Accordingly, we reverse the Examiner's decision to reject claims 1, 2, 5, 7, 8, and 11 as anticipated by Aventail Connect v3.01.

c) Claim 13

Regarding claim 13, as discussed above, claim 13 does not require an encrypted channel, but rather only requires the channel to be "secure." Aventail Connect v3.01 discloses that Aventail Connect is a "secure proxy client." Aventail Connect v3.01, 1. Accordingly, even though decryption occurs at Aventail Connect, the channel is still secure as required in claim 13. *See* RAN 11.

As to Patent Owner's argument that Aventail Connect v3.01 does not disclose a "direct" channel that is encrypted between an application and a remote host, we are not persuaded. That is, Patent Owner acknowledges the '151 Patent Specification "discloses that the communication traverses a network (or networks) through which it is simply passed or routed via various network devices such as Internet Service Providers, firewalls, and routers" and that this constitutes "'direct' communication." PO Appeal Br. 16, citing '151 Patent Figs. 2, 24, 28, 29, 33. Patent Owner's basis for contending Aventail Connect v3.01 does not disclose a "direct" channel is that in Aventail Connect v3.01, the encrypted communications channel passes through the SOCKS server, "which may create a new connection to the remote host." *Id.* at 16–17. Patent Owner alleges that "[t]his operation is different from a regular firewall or an edge router, where the traffic is destined for the target device and not for the firewall or the edge router." *Id.* at 17, citing Aventail Connect v3.01 6, 7, 12.

We observe that the Federal Circuit held with respect to U.S. Patent 6,501,135 (the "'135 Patent"), which shares a common specification with the '151 Patent, that the term "VPN between the client computer and the target computer" requires a "direct communication between the client computer and target computer" as a result of prosecution-history disclaimer. *VirnetX Inc. v. Mangrove Partners Master Fund Ltd.*, 778 F App'x at 909–910. However, the Federal Circuit did not indicate that a "direct" channel was required in the '151 Patent. *Id.* at 911 (holding "the Board erred in construing claims 1, 3–4, 7–8, 10 and 12 of the '135 Patent"). We have also not been directed to a specific definition of "direct" in the '151 Patent.

Even if the claims of the '151 Patent do require a "direct" channel, Patent Owner has not sufficiently explained how such a requirement in the claims necessarily distinguishes Aventail Connect v3.01. Although Patent Owner has referred to Internet Service Providers, firewalls, and routers as being examples of intervening devices allowed in a direct communication, given the claim language in claim 13, we do not find such examples necessarily distinguish the SOCKS server in Aventail Connect v3.01 even though the SOCKS server is disclosed as being "more than a standard security firewall." PO Appeal Br. 17; Aventail Connect v3.01 7.

The '151 Patent discloses the secure channel may be established preferably by utilizing a hopping scheme that is "preferably performed transparently to the user." '151 Patent, col. 38, ll. 62–66. The hopping scheme involves the use of Tunneled Agile Routing Protocol (TARP) packets and routers, which instead of indicating a final destination in the destination field of the IP header, points to the next-hop in a series of TARP router hops. *See* '151 Patent, col. 3, ll. 8–27; *see also* col. 31, ll. 25–42 (describing the use of edge routers and a plurality of Internet Service Providers). Aventail Connect v3.01 discloses Aventail Connect "is designed to run transparently on each workstation" and discloses the SOCKS server sends traffic to the Internet or an external network. Aventail Connect v3.01 7. We simply do not see how a requirement that the secure channel be "direct" distinguishes between the claim language in claim 13, which does not require a particular protocol or network layer for the direct connection and the disclosure in Aventail Connect v3.01.

Accordingly, we affirm the Examiner's decision to reject claim 13 as anticipated by Aventail Connect v3.01.

d) Claim 14

Regarding claim 14, the Examiner found Aventail Connect v3.01 discloses sending a proxy request to a SOCKS server, which depending on the security policy for that request, would correspond to a request for an encrypted channel between the secure server and a client. RAN 13–14, citing Aventail Connect v3.01 12.

Patent Owner contends Aventail Connect v3.01 fails to disclose "sending a request to the secure server to establish a secure channel between the secure server and the client" because the Aventail Connect software sends a proxy request to the SOCKS server, where the encryption module may not be enabled and selected by the SOCKS server, such that the proxy request is not a request to establish a secure channel. PO Appeal Br. 18–19.

Cisco contends the claims require only "sending a request" to establish an encrypted channel, and while some proxy requests in Aventail Connect v3.01 may disable encryption, such does not take away from the disclosure that certain proxy requests establish an encrypted channel. Cisco Resp't Br. 9.

As discussed above, the channel established in Aventail Connect v3.01 is "secure" as recited in claim 13. We agree with the Examiner and Cisco that Aventail Connect v3.01 discloses sending a request to a secure server as recited in claim 14. Aventail Connect v3.01, 12; *see id.* at 7. Patent Owner's contention that in some instances, the request may not be for a secure channel, does not change the disclosure in Aventail Connect v3.01 that other proxy requests are for secure channels.

Accordingly, we affirm the Examiner's decision to reject claim 14 as anticipated by Aventail Connect v3.01.

D. Anticipation – AutoSOCKS (Issue 2)

For similar reasons as discussed above for Aventail Connect v3.01, Patent Owner contends AutoSOCKS has not been shown to be publically accessible. PO Appeal Br. 5–7. We are not persuaded by Patent Owner's arguments for similar reasons as discussed above with respect to Aventail Connect v3.01, because the Chester Declaration provides similar statements with respect to AutoSOCKS as provided for Aventail Connect v3.01. *See* Chester Decl. ¶¶ 10–13, 18–19.

There is no dispute that the disclosure of AutoSOCKS is "substantially similar" to Aventail Connect v3.01. RAN 15–16; PO Appeal Br. 21; *see* Ans. 2–22.

Accordingly, we affirm the Examiner's rejection of claims 13 and 14 and reverse the Examiner's rejection of claims 1, 2, 5, 7, 8, and 11 for similar reasons as discussed above for Aventail Connect v3.01.

E. Obviousness – Aziz and Edwards (Issue 12)

1. The Examiner's Findings

The Examiner found Aziz discloses a DNS proxy module in the form of resolver 225. More specifically, the:

> examiner submits that Aziz teaches or suggests 'a domain name server (DNS) proxy module . . . because 'resolver 225 could follow the referral chain to the name server for the domain of inside host 140 or could pass the query on to local NS 250.' . . . Thus, it would have been obvious to those of ordinary skill in the art that the resolver of Aziz represents a DNS proxy module.

Ans. 94–95, citing Aziz, col. 6, l. 62 – col. 7, l. 7; col. 10, ll. 36–42. As the Examiner further explained, the operation of resolver 225 parallels the operation of the name server 120 in that the resolver 225 receives a query from application 215 and sends a query to the name server 120, and when the name server 120 sends a response back to the resolver 225, the resolver

225 checks for "an SX record"[21] in the response. ACP 68, citing Aziz, col. 9, ll. 54–56, col. 10, ll. 4–5, 36–39, 42–48, Figs. 3, 4A.

The Examiner found Edwards generally teaches technologies for securing HTTP web servers and related applications to prevent unauthorized access to data. Ans. 96. The Examiner found Edwards discloses a DNS proxy module (an object gateway), and the object gateway intercepts a name request sent by a client using a naming interceptor. *Id.* at 99–100. The Examiner found that Edwards discloses the object gateway includes both the proxy and the naming interceptor and determined it would have been obvious to modify the name server software of Aziz to additionally intercept name service requests, as taught by Edwards. *Id.* at 101.

The Examiner found Aziz discloses determining whether a DNS request corresponds to a secure server by checking whether the record is an associated "secure exchanger" or "SX" record. *Id.* at 104. The Examiner found Edwards discloses determining whether an intercepted name request specifies an available target, where available targets have enabled authentication and authorization, and where an administrator can adjust the access control as required. *Id.* at 104–105, citing Edwards 933. The Examiner determined it would have been obvious to one of ordinary skill in the art that a target with authentication and authorization enabled corresponds to a secure server. *Id.* at 105.

The Examiner found Aziz discloses automatically initiating an encrypted channel between the client computer and the target computer by

[21] Aziz indicates "a new resource record type" is "herein called an SX record" that responds "to requests for information needed for secure communications with protected hosts in that domain." Aziz, col. 4, ll. 9–12.

creating a tunnel map entry, which is used to encrypt messages, and after creating a tunnel map entry the application can communicate securely with the inside host 140. *Id.* at 111. The Examiner found that Aziz's disclosure of keeping secure message information up-to-date without relying on human intervention as support that Aziz discloses initiating the virtual private network "automatically." *Id.* The Examiner found Edwards discloses a domain name server proxy module in its disclosed object gateway, which intercepts name service requests. *Id.* at 99–101, citing Edwards 932, 934, Figs. 4, 6.

The Examiner determined it would have been obvious to have combined the Aziz network structure with the additional techniques disclosed in Edwards because the combination utilizes known techniques as disclosed in Edwards to improve a similar system as disclosed in Aziz in the same way. *Id.* Specifically, the Examiner determined combining the transparent encryption of Aziz with the interception of name service requests as taught by Edwards "allow[s] the Aziz network to provide its transparent encryption services with little or no client configuration required[, which] would further improve the transparency and ease of deploying the Aziz architecture." *Id.* The Examiner determined also that combining Aziz's transparent encryption with Edward's name service requests is a combination of known methods that merely produces a predictable result, such as providing transparent packet encryption in response to intercepted service requests. *Id.* at 96.

2. *Patent Owner's Contentions*

Patent Owner contends that although the Examiner identifies the resolver 225, which is located in authorized client 210 disclosed in Aziz as the DNS proxy module, the rejection cites to outside NS 120, separate from authorized client 210, as performing recited features of the DNS proxy module recited in claim 1. PO Appeal Br. 50. Patent Owner contends that any proposed modification of Aziz to locate resolver 225 outside NS (name server) 120 and outside of client 210 would be improper, and would render Aziz unsatisfactory for its intended purpose. *Id.* at 50–51. Patent Owner contends Edwards does not disclose or suggest the object gateway disclosed therein performs all of the features of the DNS proxy module recited in claim 1. *Id.* at 51. Patent Owner contends neither the Office nor Cisco explains how it would have been obvious to have combined the object gateway disclosed in Edwards with outside NS 120 and resolver 225 disclosed in Aziz. *Id.* at 51–52. Patent Owner contends also that both Aziz and Edwards disclose receiving, but not intercepting DNS requests, and further that it would not have been obvious to have modified Aziz to intercept DNS requests. *Id.* at 52–54. Patent Owner argues that Aziz does not disclose or suggest determining whether the intercepted DNS request corresponds to a secure server, because checking for an SX record is not the same as determining whether a DNS request corresponds to a secure server. *Id.* at 54–56. Patent Owner contends Edwards does not suggest determining whether the DNS request corresponds to a secure server. *Id.* at 56. Patent Owner argues the combination of Aziz and Edwards fails to disclose or suggest automatically initiating an encrypted channel between the client and the secure server. *Id.* at 57. In particular, Patent Owner argues Aziz's disclosure of creating a tunnel map entry does not automatically initiate an encrypted channel, and Edwards does not make up for this deficiency. *Id.* at 57–59.

3. *Cisco's Contentions*

Cisco contends Patent Owner has misconstrued the Examiner's position and the Examiner found the Aziz's resolver 225 discloses all the recited features. Cisco Resp't Br. 22. As to a DNS module that intercepts DNS requests sent by a client, Cisco contends that even under Patent Owner's interpretation of "intercept," Edwards discloses the recited limitation. *Id.* at 22–23. Cisco contends the SX records disclosed in Aziz contain an identifier of a secure exchanger, and as a result, Aziz discloses determining whether the corresponding host supports secure communications. *Id.* at 23. Cisco argues that Aziz discloses encrypted messages are sent from the host computer to the client without human intervention, and as such, are automatically initiated. *Id.* at 23–24.

4. *Analysis*

We limit our discussion to claim 1, which is sufficient to resolve the issues associated with this rejection. *See* 37 C.F.R. § 41.67(c)(vii). We are not persuaded by Patent Owner's arguments for the reasons that follow.

a) *DNS Proxy Module*

Aziz discloses an application 215 running on authorized client 210 (client) makes a query (DNS request) for the address of an inside host 140, which is received by a resolver 225 (DNS proxy module), and can follow a referral chain to the name server for the domain of inside host 140. Aziz, col. 10, ll. 35–41, col. 8, ll. 19–25; Figs. 1, 2A. Aziz also discloses an inside host 140 may be in protected zone 180 (secure server). *Id.* at col. 10, ll. 28–30, Fig. 1. The resolver 225 receives a response to the query checks to see if there is an "SX record," an identifier indicating there is a "secure exchanger" associated with the host name (the DNS proxy module determines whether the intercepted DNS request corresponds to a secure server as recited in claim 1). Aziz, col. 6, ll. 27–29, col. 10, ll. 42–43; Figs. 2A, 4A. Aziz discloses "[a] secure exchanger is a machine that handles secure communications for itself or for another machine (e.g. performs encryption or decryption)." Aziz, col. 6, ll. 29–33. Aziz discloses an example of a secure exchanger is firewall 110. Aziz, col. 6, ll. 36–38. Aziz discloses resolver 225 creates a "tunnel map entry," which "provides all the information that crypto-processor 230 needs to encrypt messages to inside host 140" to allow for secure communications (when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server as recited in claim 1). Aziz, col. 11, ll. 55–62; *see also id.,* col. 12, ll. 50–56 (disclosing the query is encrypted).

Consistent with the above discussion of Aziz, the Examiner found the resolver 225 of Aziz is the DNS proxy module, and not the combination of the resolver 225 and the outside NS 120 as asserted by Patent Owner. ACP 68; RAN 42–43; Ans. 94–95. Thus, contrary to Patent Owner's contentions, the Examiner does not rely on the combination of the resolver and the outside NS 120 as the DNS proxy module. Patent Owner's arguments focus on the joint operation of the resolver 225 with outside NS 120 and why relocating the resolver 225 would not have been obvious, without explaining what particular operation of the DNS proxy server the resolver 225 fails to perform.

b) "Intercepts DNS Requests"

We are not persuaded by Patent Owner's argument that Aziz does not disclose intercepting DNS requests. PO Appeal Br. 53. We agree with

Cisco's contention, adopted by the Examiner, that Patent Owner has not provided a sufficient explanation as to why the broadest reasonable interpretation of "intercepting" does not include receiving requests as taught by Aziz. ACP 67–68. In this regard, Patent Owner argues Cisco's construction of "intercepting" as reading on receiving a request disclosed in Aziz is inconsistent with a related proceeding where the Board allegedly determined the term "intercepting a request" "would require 'receiving and acting on' a request, the request being 'intended for' receipt at destination other than the destination at which the request is intercepted." PO Appeal Br. 53–54, quoting *Apple Inc. v. VirnetX Inc.,* IPR2014-00237, Paper No. 15 at 12 (PTAB May 14, 2014) (Decision on Institution). We observe, however, that the interpretation of "intercepting a request" as set forth by the Board under the broadest reasonable construction standard was "receiving a request pertaining to a first entity at another entity." *Apple Inc. v. VirnetX Inc.*, IPR2014-00237, Paper No. 41 at 12 (PTAB May 11, 2015) (Final Written Decision). Although this Decision was appealed, the Federal Circuit did not decide the appeal related to IPR2014-00237. *VirnetX Inc. v. Apple Inc.*, 665 F. App'x 880, 881 (Fed. Cir. 2016) ("We affirm, resolving the subject appeals on the grounds discussed by the PTAB in *VirnetX II* [IPR2014-00238].") In addition, the portions of the Specification of the patent discussed in IPR2014-00237 (US 8,504,697 B2 "the '697 Patent") are the same as in the '151 Patent. *Compare* '697 Patent, col. 40, ll. 31–33, Fig. 26, *with* '151 Patent, col. 37, ll. 60–62, Fig. 26. Thus, we do not agree with Patent Owner that the construction proposed by Cisco is inconsistent with IPR2014-00237.

c) Steps (i) and (iii)

As pointed out by the Examiner, Aziz discloses the data field of the SX record contains "the identifier (e.g., name or address) of a 'secure exchanger' associated with the owner of the record," where the "secure exchanger is a machine that handles secure communications for itself or for another machine (e.g., preforms encryption or decryption)." Aziz, col. 6, ll. 27–32, col. 10, ll. 42–52. Thus, in contrast to Patent Owner's arguments, by checking if the request contains an SX record, resolver 225 determines whether the request corresponds to a secure server. Indeed, the '151 Patent discloses that determining whether a secure site has been requested is through "domain name extension, or by reference to an internal table of such sites." '151 Patent, col. 37, ll. 62–66.

We are not persuaded by Patent Owner's argument it would not have been obvious to have modified Aziz to intercept name service requests, because there is an inadequate explanation as to how Edwards's interception techniques would provide transparent encryption services with little or no client configuration, where Aziz already achieves this goal. PO Appeal Br. 53, citing Aziz col. 3, ll. 3–12; Keromytis Decl. ¶ 163.

We observe that the Examiner as well as Cisco, rely on the position that Aziz discloses receiving a DNS request, and Edwards discloses it is well known in the art for an intermediary module such as Aziz's resolver 225 to intercept requests. *See* ACP 67, citing Cisco's comments filed on August 17, 2012, 32–33. Thus, Edwards merely provides additional support that it is well known for DNS requests to be intercepted and directed to appropriate modules for resolution. *See* Edwards 934–936 (disclosing the "proxy is configured to use the naming interceptor as its naming service" and "the naming interceptor controls whether or not a service is available to any external clients."). Moreover, Edwards discloses that there is a choice as to the placement of the object gateway, which includes the proxy and interceptors, such as on a remote machine, and also that the object gateway can share processes with many other gateway objects. Edwards, 931, 936; Fig. 6.

Patent Owner's arguments consider each reference individually rather than what one of ordinary skill in the art would have understood from the collective teachings of the references.

Thus, for these reasons, we affirm the Examiner's rejection of claims 1, 7, and 13.

F. Obviousness – Claims 5 and 11 Aziz, Edwards, and Martin (Issue 13)

Status of Martin as Prior Art

Patent Owner contends Cisco provided no evidence that Martin was publically accessible more than one year prior to the alleged earliest effective date, because there is no indication what the date of February 21, 1998 on the first page of Martin means. PO Appeal Br. 9. Patent Owner contends the additional evidence submitted by Cisco showing public accessibility should not be considered because it was not submitted with the Request. *Id.*, citing 37 C.F.R. §§ 1.947–1.948.

Cisco contends that additional evidence of record established Martin was available to the public at least as early as February 21, 1998, and the citation to such evidence is proper. Cisco Resp't Br. 7–8, Exs. I–K, and citing the Manual of Patent Examining Procedure (MPEP) § 716.01(c).

The Examiner determined Martin was a publically accessible technical report from Boston University's Computer Science Department, posted at http://www.cs.bu.edu/techreports/pdf/1997-022-lanon.pdf. ACP 8. The Examiner's conclusion is also supported by the evidence cited in Cisco's Respondent Brief, showing that Martin was catalogued with a publication date of February 21, 1998. Cisco Resp't Br. Exs., H–J.

Patent Owner does not substantively challenge the Examiner's position that Martin is prior art based on the public availability of Martin from Boston University's Computer Science Department. We agree with Cisco that Patent Owner's contentions that 37 C.F.R. §§ 1.947 and 1.948 do not allow for additional evidence are unsupported. In this instance, Patent Owner challenged the position that Martin was prior art, and in response Cisco filed additional evidence supporting that position. *See* MPEP § 2166.05(II). We do not see how Cisco's evidence would not be allowed in this situation.[22]

[22] *See* Petition Decision January 1, 2016, dismissing PATENT OWNER'S PETITION TO REOPEN PROSECUTION filed November 23, 2015 and determining that the Examiner's rejections, including those involving Martin, were not altered by the Examiner's reliance on Cisco's exhibits filed with Cisco's written comments submitted on August 12, 2012 and the online post at http://www.cs.bu.edu/techreports/pdf/1997-022-lanon.pdf, and thus did not constitute new grounds of rejection. DECISION ON PETITION entered January 20, 2016, 4–6. Although Cisco's written comments, Patent Owner's petition, and the Petition Decision all refer to "exhibits M, N, and O," our review of the record indicates that the exhibits at issue are actually Exhibits H, I, and K as attached to Cisco's Respondent Brief, with Cisco's written comments incorrectly referring to those exhibits as "exhibits M, N, and O," an error which carried through Patent Owner's petition and the Petition Decision without being identified in those documents. *See* "COMMENTS BY THIRD PARTY REQUESTER PURSUANT TO 37 C.F.R. § 1.947" filed by Cisco on August 17, 2012, at v, 4–5.

Accordingly, we agree with the Examiner that Martin constitutes publically accessible prior art.

1. Claims 5 and 11

Substantively, Patent Owner contends that claims 5 and 11 depend from claims 1 and 7, respectively and Martin does not remedy the deficiencies of Aziz and Edwards with respect to the independent claims. PO Appeal Br. 60. Accordingly, we affirm the Examiner's rejection of claims 5 and 11 as obvious over Aziz, Edwards, and Martin for similar reasons as discussed above for claims 1 and 7.

G. Obviousness – Kiuchi and Edwards (Issue 14)

We limit our discussion to claims 1, and 13–16, which is sufficient to resolve the issues associated with this rejection. *See* 37 C.F.R. § 41.67(c)(vii).

Similar to the rejection of the claims over Aziz and Edwards discussed above, the Examiner cited Edwards for the concept of incorporating naming interceptors into Kiuchi's client-side and server-side proxies in rejecting claims 1–4, 6–10, and 12–16. Ans. 136, 140–141.

Patent Owner contends Kiuchi teaches away from the features of claim 1 as discussed above for the anticipation rejection. PO Appeal Br. 61–64. Patent Owner contends Edwards does not make up for such deficiencies in Kiuchi. *See id.* at 61–62.

Cisco contends Kiuchi discloses the limitations of claim 1 and Edwards shows that it was well-known to intercept requests sent to name services, and as a result, the Examiner's rejections should be affirmed. Cisco Resp't Br. 24–26.

As discussed above, because Kiuchi only discloses encrypted communications between the client-side and server-side proxies and not between the client and secure server, we reverse the Examiner's rejection of claims 1–4, 6–10, and 12 as obvious over Kiuchi and Edwards.

However, because, as discussed above, claim 13 only requires establishing a "secure" channel between the client and secure server, we affirm the Examiner's decision to reject claim 13 as obvious over Kiuchi and Edwards.

1. Claim 14

Patent Owner presents separate arguments for dependent claim 14. PO Appeal Br. 66–67. In this regard, Patent Owner relies on the position that Kiuchi fails to disclose determining whether the client is authorized to access the secure server, which we found unpersuasive above. *Id.* at 66. Patent Owner argues additionally that the Examiner and Cisco no longer rely on additional disclosures in Edwards for the recitations in claim 14. *Id.* at 66–67.

We agree with Patent Owner, that the Examiner does not rely on any additional rationale that Edwards discloses checking a client's authorization to access a secure server. RAN 39. Thus, we affirm the Examiner's rejection of claim 14 based on the above discussion that Kiuchi discloses determining whether the client is authorized to access the secure server.

2. Claims 15 and 16

Patent Owner contends Kiuchi fails to disclose returning a "host unknown" error message to a client as recited in claim 15 and as discussed above. PO Appeal Br. 67–68. Patent Owner argues also that Edwards does not make up for the deficiencies in Kiuchi. *Id.* at 68–69. In particular, Patent Owner contends the "object not found" error disclosed in Edwards, generated when a requested name "is not in the list of available targets" even when combined with Kiuchi, would not result in returning a "host unknown" error message to the client. *Id.* at 68; Ans. 153–154; 165.

The Examiner and Cisco contends that Patent Owner's arguments are identical to the arguments regarding Kiuchi alone. RAN 40; Cisco Resp't Br. 26.

For the reasons discussed above, we agree with Patent Owner that Kiuchi does not disclose returning a "host unknown" error to the client. Further, we agree with Patent Owner that the Examiner and Cisco fail to sufficiently explain how the combination of the "object not found" error in Edwards would have remedied the deficiency identified above in Kiuchi.

As a result, we reverse the Examiner's decision to reject claim 15 as obvious over Kiuchi and Edwards.

Because claim 16 depends from claim 15, we reverse the Examiner's rejection of claim 16 as well.

In sum, we affirm the Examiner's decision to reject claims 13 and 14 as obvious over Kiuchi and Edwards. However, we reverse the Examiner's decision to reject claims 1–4, 6–12, 15, and 16 as obvious over Kiuchi and Edwards.

H. Obviousness – Claims 5 and 11 over Kiuchi, Edwards, and Martin (Issue 15)

Claims 5 and 11 depend from claims 1 and 7, respectively, and recite the step of initiating encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server. Martin is cited for disclosing an IP address hopping scheme that can co-exist with the system of Kiuchi in view of Edwards. Ans. 168–169. Martin fails to cure the deficiencies identified in Kiuchi and Edwards for claims 1 and 7. As a result, we reverse the Examiner's decision to reject claims 5 and 11 for similar reasons as discussed above with respect to claims 1 and 7.

I. Obviousness – Beser and Kent (Issue 4)

1. The Examiner's Findings

The Examiner found Beser discloses a process where an IP tunnel, and encrypted channel, is established between two network devices through a third trusted network device on a public network, corresponding to a DNS proxy module as recited in claim 1. Ans. 25, citing Beser, col. 10, ll. 37–41, col. 11, ll. 32–36. The Examiner found that Beser, in an IP telephony example, discloses a domain name may be used to determine if an encrypted channel needs to be established securely, and if so, the trusted third party network device will negotiate with the first and second network devices to establish an encrypted channel between the first and second network devices. *Id.,* citing Beser, Fig. 4, col. 11, ll. 9–44, col. 11. l. 58 – col. 12, l. 19. The Examiner found Beser discloses VPNs (virtual private networks) are routinely implemented for IP tunnels and IPsec is a known technique that can be used to encrypt communications within those types of encrypted channels. *Id.* at 25–26, citing Beser, col. 1, ll. 54–57, col. 2, ll. 7–13. The Examiner found Beser discloses authentication and encryption should be used to establish IP tunnels, but Beser does not expressly require all communications between the first and second network devices to be encrypted after the IP tunnel is established, disclosing instead that in certain types of applications, the high volume of data to be transferred would make encryption of all IP packets impractical. *Id.* at 26, citing Beser, col. 11, ll. 22–24, col. 1, l. 58 – col. 2, l. 15. The Examiner found that Beser discloses IP tunneling methods ordinarily will encrypt all traffic sent between the nodes of the tunnel, and that the IPsec protocol is to be used for these encrypted IP tunnels. *Id.*, citing Beser, col. 1, ll. 54–56. Thus, the Examiner found one of ordinary skill in the art would have recognized from Beser that all communications within IP tunnel, both to initiate the tunnel and following establishment of the IP tunnel, should be encrypted other than for certain high traffic applications, and that IPsec protocol should be used to handle the encryption of the traffic being sent through the IP tunnel. *Id.*, citing Beser, col. 1, l. 54 – col. 2, l. 15. The Examiner found Beser discloses the trusted-third-party network device is a domain name server that evaluates domain names, and will take additional actions to establish the IP tunnel based on the results of the evaluation. *Id.* at 26–28, citing Beser, col. 10, ll. 37–41, col. 11, ll. 32–36, 45–49. The Examiner found that if the destination associated with the domain does not require establishment of an IP tunnel negotiated by the trusted-third-party network device, the trusted-third-party network device, as a DNS server, will return the IP address associated with the non-secure domain. *Id.* at 27–28, citing RFC 1034 at 21.[23] As to the recitation in claim 1 of "automatically initiating an encrypted channel between the client and the secure server," the Examiner found Beser discloses the trusted-third-party network device automatically negotiates with first and second devices to establish an IP tunnel therebetween. *Id.* at

[23] P. Mockapetris, "Domain Names – Concepts and Facilities," RFC 1034, October 1987. Cisco Resp't Br., Ex. D7.

28–29, citing Beser Fig. 4, col. 11, ll. 9–25, col. 12, ll. 6–19, 28–37. The Examiner found Kent discloses IPsec protocol to establish VPNs by IP tunneling, where the IPsec protocol calls for automatic encryption of all IP traffic being sent between nodes of the VPN network. *Id.* at 29–30, citing Kent 8, 13, 29–34. The Examiner determined a person of ordinary skill in the art would have relied on Kent to modify the design of Beser to incorporate IPsec to encrypt all traffic being sent in an encrypted channel between a first and a second network device in the IP tunneling procedures disclosed in Beser, rather than encrypting only traffic used to establish the encrypted channel. *Id.* at 30.

2. Patent Owner's Position

Patent Owner contends Kent has not been shown to be a printed publication because although Kent discloses "November 1998" on the first page, there is no indication Kent was publically available in November 1998. PO Appeal Br. 7–8.

Substantively, Patent Owner argues one of ordinary skill in the art would not have combined Beser and Kent because Beser teaches away from the IPsec protocol disclosed in Kent. PO Appeal Br. 23–25. Patent Owner contends Beser does not disclose a DNS proxy module or a DNS proxy module that intercepts a DNS request by a client. *Id.* at 25–27. Patent Owner contends that the Examiner's position Beser discloses forwarding an intercepted DNS request to a DNS function when the intercepted DNS does not correspond to a secure server is not supported. *Id.* at 27–28. Patent Owner argues Beser does not disclose encryption between a client and a secure server, and Beser's disclosure of encryption relates to preliminary communications between the client and DNS proxy module, which occurs before any tunneling is established. *Id.* at 28–29.

3. *Cisco's Contentions*

Cisco contends Kent is self-dated as being available as of November 1998 and states that "[d]istribution of this document is unlimited." Cisco Resp't Br. 6. Cisco argues also Kent is an early Internet Draft by the Internet Engineering Task Force (IETF), and that the IETF's process for publishing Internet Drafts indicates that such are "'replicated on a number of Internet hosts' and 'readily available to a wide audience.'" *Id.*, quoting Ex. F at 8. In addition, Cisco contends that Dr. Keromytis, Patent Owner's expert, has repeatedly cited to Kent in his own peer-reviewed papers with a date of November 1998. *Id.* at 6–7, citing Ex. G.

Cisco argues Patent Owner is precluded from arguing Beser does not disclose a DNS proxy module under 37 C.F.R. § 42.73, because the Board considered the same argument in IPR2014-00237 and determined Beser discloses intercepting a request to look up an IP address of a network device based on a domain name. Cisco Resp't Br. 11–12, citing Ex. P, Paper 41 at 28. Even so, Cisco argues the Examiner correctly concluded Beser's VoIP (Voice over Internet Protocol) initiation request is a "DNS request." *Id.* at 12. Cisco argues Beser discloses "intercepting" a request as the Board found in IPR2014-00237. *Id.* citing Ex. P, Paper 41 at 28. Cisco argues that because Beser discloses the trusted-third-party server receives different types of requests that require different responses, it would have been obvious that Beser receives different types of requests that require different responses, where some requests require initiating an encrypted channel, and others do not require an encrypted channel. *Id.* at 12–13. Cisco argues

Patent Owner has not identified any substantive teaching of the "initiating" step that is not disclosed by Beser and Kent. *Id.* at 13.

4. *Analysis*

We limit our discussion to claims 1, 2, and 5, which is sufficient to resolve the issues associated with this rejection. *See* 37 C.F.R. § 41.67(c)(vii).

a) Prior Art Status of Kent

The Examiner agreed with Cisco's position that Kent was distributed and publically accessible before February 15, 2000. ACP 7.

The content of Kent is consistent with the publication process described by RFC 2026 including the date "November 1998" on the top right corner of the first page. Kent 1. The title itself, "Request for Comments," in addition to the evidence in RFC 2026 that each RFC document was made widely available "from a number of Internet hosts," constitutes sufficient evidence that Kent was intended for publication and would have been accessible to interested artisans seeking documents related to "Internet standards." Kent 1; Cisco Resp't Br., Ex. F, 6, 8. Accordingly, we find that Kent was publically accessible as of February 15, 2000.[24]

b) Teaching Away

We are not persuaded that Beser teaches away from incorporating IPsec protocol as disclosed in Kent. We agree with the Examiner's position that Beser's disclosure relates to certain limitations that with respect to the

[24] We observe that the Federal Circuit has previously determined that Kent is prior art and that VirnetX is collaterally estopped from arguing whether Kent is a printed publication. *VirnetX Inc. v. Apple, Inc.*, Appeal No. 2017-2490, 2017-2494, slip op. 2, n.1, 4 (Fed. Cir. Dec. 10, 2018).

use of encrypted packets with VoIP, including the feasibility of encrypting packets at the source and decrypting packets at a destination for certain data formats, and the amount of computing power required to encrypt or decrypt IP packets on the fly and does not amount to a teaching away. *See* Beser, col. 1, l. 58 – col. 2, l. 35. Indeed, Beser discloses the use of encryption for IP packets to ensure that the unique identifier cannot be read on the public network. Beser, col. 11, ll. 22–25, col. 20, ll. 11–14. As discussed above, Kent discloses IPsec protocol to establish VPNs by IP tunneling, where the IPsec protocol calls for automatic encryption of all IP traffic being sent between nodes of the VPN network. Kent 8, 13, 29–34; *see also id.* at 7 (disclosing IPsec allows the creation of an encrypted tunnel between each pair of hosts communicating across two security gateways). Accordingly, Beser contemplates the use of encryption in the methods disclosed in Kent, and as such, Patent Owner's argument that Beser teaches away from encryption is not persuasive.

c) Steps (i) and (ii)

We are also not persuaded by Patent Owner's argument that Beser fails to disclose that the trusted-third-party network may function as a DNS proxy module that performs the step of: "when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer" as recited in claim 1. Beser expressly discloses the trusted-third-party network device is a domain name server. Beser, col. 11, ll. 32–36. Patent Owner points out that Beser discloses that the originating and terminating security of the connection is determined based on the private address of the requesting device 24 and the terminating device 26. Appeal Br. 22–23, citing Beser, Fig. 1, col. 2, ll. 36–39, col. 7, l. 62 – col. 8, l. 4, col. 10, ll. 2–6, col. 11, ll. 9–10, 26–32, col. 11, l. 59 – col. 12, l, 54. However, Patent Owner does not meaningfully address the Examiner's rationale discussed above that because Beser discloses domain servers, when the requesting device requests a connection to a non-secure server, such a request would be forwarded to a DNS function that would return an address of a nonsecure computer as evidenced by RFC 1034. Ans. 27. Accordingly, we are not persuaded by Patent Owner's contentions that the Examiner filled in missing limitations simply because they are not prohibited by Beser. PO Reb. Br. 7–8.

d) Claims 2, 8, and 14

Claim 2, which is representative, depends from claim 1, and recites, in pertinent part, "wherein step (iii) comprises the steps of: (a) determining whether the client is authorized to access the secure server." The Examiner found either Beser inherently discloses the limitation or Kent discloses IPsec involves an authentication step, rendering claim 2 obvious. Ans. 30–31; RAN 25, citing Kent 4, 47.[25]

Patent Owner contends that the cited portions of Beser fail to disclose requiring authenticating a client computer in conjunction with a tunneling association, and Beser discloses no reason for requiring the client be authorized or authenticated in accessing the third party network device of Beser or the secure server. PO Appeal Br. 31. Patent Owner acknowledges Beser requires authentication, but argues authentication is not the same as authorization. *Id.* Patent Owner contends the IPsec access control disclosed

[25] The cited portion of Kent attributed to page 47 appears to us to be on page 45.

in Kent does not make up for the deficiencies of Beser, because Kent is not concerned with whether a client is authorized to access a secure server. *Id.* at 32.

Cisco contends Kent discloses a variety of security services including "access control" such that one of ordinary skill in the art would have found controlling access of a client would have rendered obvious the determining whether the client is authorized as claimed. Cisco Resp't Br. 13–14, citing ACP 43; Kent 4.

We are not persuaded by Patent Owner's arguments. As discussed above, the combination of Beser and Kent discloses establishing an encrypted channel between a client and a secure server using the IPsec protocol disclosed in Kent. Beser discloses the IP packets may require authentication (Beser, col. 11, ll. 22–25), and Kent discloses access control, which is "prevention of use of a resource in an unauthorized manner." Kent 4, 45. While "authentication" and "access control" may encompass different concepts (*see* Kent 45), such does not undermine the Examiner's position that in applying the IPsec protocol of Kent to Beser, one of ordinary skill in the art would have included access control in order to determine whether the client is authorized to access the secure server as recited in claim 2. The Examiner's position is consistent with Beser's disclosure of authentication in order to ensure the security of the packets that are communicated between an originating device and a terminating device.

e) Claims 5 and 11

As discussed above, claim 5, which is representative, depends from claim 1 and recites, in pertinent part, an IP hopping scheme. The Examiner found Beser discloses Network Address Translator (NAT) protocol, which is an IP hopping scheme as recited in claim 5. Ans. 35, citing Beser, col. 2, ll. 18–27; ACP 47.

Patent Owner contends that the Examiner's position the NAT protocol disclosed in Beser satisfies the "IP hopping scheme" recited in claim 5 is conclusory and without any support. PO Appeal Br. 33. In addition, Patent Owner contends Beser teaches against the use of NAT, disclosing NAT only to show it is not to be used with VoIP applications, the technology with Beser is primarily concerned. *Id.* at 33–34.

Cisco contends the NAT protocol acts as an interface between a local network and a global network, which works by changing the originating IP address in packets before forwarding the packets to an outside server, and is an IP address hopping scheme within the meaning of claims 5 and 11. Cisco Resp't Br. 14, citing Ex. D1355 (RFC 1631). Cisco additionally contends Beser states only that NAT "may be" inappropriate for the transmission of multimedia or VoIP due to computer power limitations, and that nothing in Beser states that NAT should not be used or cannot be used in any tunneling associations, which is not a teaching away from NAT in all applications. *Id.*

We are not persuaded by Patent Owner's arguments. In particular, Patent Owner has not sufficiently explained why there is insufficient support for the Examiner's position that the NAT protocol disclosed in Beser is an IP hopping scheme as recited in the claims. Beser discloses "[a]nother method for tunneling is network address translation [(NAT)]." Beser 2, ll. 18–22. As pointed out by Cisco, the NAT protocol is understood by those skilled in the art as changing the originating IP address packets before forwarding packets to an outside server as disclosed in RFC 1631, which constitutes an

IP hopping scheme within the meaning of claims 5 and 11. Ex. D1355 (RFC 1631) 2–5.

In addition, we agree with the Examiner and Cisco that Beser does not necessarily teach away from the use of NAT protocol in all applications. Although it is true Beser discloses a number of limitations in using the NAT protocol, disclosing that use of the NAT protocol is "computationally expensive," prevents certain types of encryption from being used, is not compatible with a number of existing applications, and due to computer power limitations, "may be" inappropriate for the transmission of multimedia of VOIP packets (Beser, col. 2, ll. 18–35), such limitations would not prevent the use of the NAT protocol when these considerations were not at issue.

As a result, we affirm the Examiner's rejection of claims 1, 2, 5, 7, 8, 13, and 14 as obvious over Beser and Kent.

J. Objective Indicia of Non-Obviousness

"[T]he obviousness inquiry centers on whether 'the claimed invention as a whole' would have been obvious." *Rambus Inc. v. Rea,* 731 F.3d 1248, 1257–58 (Fed. Cir. 2013) (citation omitted). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.,* 383 U.S. 1, 17–18 (1966). *See also KSR Int'l Co. v. Teleflex Inc.,* 550 U.S. 398, 406–07 (2007). Additionally, "[t]he obviousness assessment depends on what the prior art teaches and on what the non-prior-art evidence of 'secondary considerations' (or objective indicia) may indicate about whether the invention would have been obvious at the relevant time." *See Institut Pasteur & Universite Pierre et Marie Curie v. Focarino,* 738 F.3d 1337, 1344 (Fed. Cir. 2013) (citing *KSR,* 550 U.S. at 406–07).

Patent Owner contends that there is evidence of objective indicia including "Long-felt Need, Failure of Others, Skepticism, Commercial Success, and Praise and Acceptance by Others" that demonstrate non-obviousness. PO Appeal Br. 71 (emphasis omitted).

Cisco contends the Board has reviewed essentially the same evidence as presented in the instant reexamination in another reexamination, and found it to be insufficient to outweigh the evidence of obviousness. Cisco Resp't Br. 27, citing Ex. R, Decision on Appeal in Reexamination Control 95/001,792 (Appeal No. 2014-000591) of US Pat. 7,188,180, 18–23.

The Examiner determined the evidence presented by Patent Owner was entitled to little weight and thus did not outweigh the evidence of obviousness on the record. RAN 53–54; ACP 84.

We agree with the Examiner and Cisco that the evidence of objective indicia does not outweigh the evidence supporting a conclusion of obviousness discussed *infra*. Our reasoning for this determination follows.

1. Long-Felt Need

Patent Owner contends evidence of record demonstrates the computer-security and internet-security industries have had a long-felt need to easily and conveniently establish secure communications links, and the inventions claimed in the '151 Patent combine both the ease of use and the security aspects of a VPN, without sacrificing one or the other. Appeal Br. 71–72, citing Short Decl.[26] ¶¶ 3, 4–9, 11; Exs. B-1, B-2, B-4.

Cisco contends Patent Owner has not provided any analysis that any claim or claim term actually recites or incorporates the notion of easily and conveniently establishing VPN communications. Cisco Resp't Br. 28.

We are of the view that Patent Owner has not provided sufficient evidence of a persistent, long felt need or failure of others for establishing encrypted or secure communication links in an easy manner. The evidence provided by Patent Owner only establishes that secure communications was an area of interest. That is, although Patent Owner points to the Short Declaration and the Exhibit thereto "Living in your own private Idaho" for support that access to secure communications in an easy manner represented long-felt need (Short Decl. ¶¶ 8–9, citing Ex. B-4, 1 (VNET00219638)), such does not support Patent Owner's position because the claimed invention does not recite any particular ease associated therewith in initiating encrypted or secure communications. In addition, the assertions by Patent Owner that the Defense Research Projects Agency (DARPA) funded programs "were focused on the need to provide easy-to-enable secure communications" by significantly funding various research programs (Short Decl. ¶ 4) is not supported by the evidence cited. In particular, the Exhibits pointed to in the Short Declaration, while directed to secure communications, do not appear to discuss any particular aspect of easily enabling such communications. Short Decl. ¶¶ 4–5; Ex. B-1,

[26] Declaration of Dr. Robert Dunham Short III, dated July 19, 2012 ("Short Decl.").

VNET00219302, 319–321; Ex. B-2, VNET00219244, 284, 298–299, 593, 625; Ex. B-3, 1 (VNET00219634). Patent Owner's reference to relationships between In-Q-Tel and SAIC and investments in technology allegedly resulting in the '151 Patent, while evidence of the commitment to the technology, does provide any particular evidence that the '151 Patent claims provide a solution to a long-felt unfulfilled need.

2. *Failure of Others*

Patent Owner contends others attempted to create easy-to-enable secure communications, which attempts failed. PO Appeal Br. 72, citing Short Decl. ¶¶ 4, 5, 10, 11; Ex. B-3 "Dynamic Coalitions" program.

Cisco contends Patent Owner has failed to explain how any of program goals from the Dynamic Coalitions program, which relate to "provid[ing] 'continuous network operations even after a cyberattack' and supporting 'distributed rather than hierarchical coalition security policies,'" relate to the claims and how any alleged failure of these programs address a need for quick and easy secure communications. Cisco Resp't Br. 28, citing Ex. B-3, 1–2.

We agree with Cisco in this regard. In addition to the discussion above with respect to long-felt need, Patent Owner's evidence does not support Patent Owner's contentions. That is, the evidence simply describes the goals of the Dynamic Coalitions program, to ensured continued communications when the composition of a coalition changes or when an ad hoc area network is attacked. Ex B-3 VNET00219634. Patent Owner does not point out with any particularity where the evidence discusses easy-to-enable secure communications, such that, even if we were to agree that the claims of the '151 Patent embody such a system, there is no indication of failure to achieve such a system from the Dynamic Coalitions program.

3. *Skepticism*

Patent Owner contends the technology of the '151 Patent was met with skepticism by those skilled in the art. PO Appeal Br. 72–73, citing Short Decl. ¶¶ 13–15.

Cisco contends Patent Owner has not provided sufficient details to support Patent Owner's arguments of skepticism. Cisco Resp't Br. 28.

We are not persuaded by the statements made in the Short Declaration that the claims of the '151 Patent were contrary to the accepted wisdom, a belief reinforced by the IT offices of many large companies and institutions. Short Decl. ¶¶ 13–14. The evidence cited by the Short Declaration does not discuss difficulties with VPN systems being not easily or conveniently enabled, but rather discusses the ability of VPNs to be used "without requiring companies to rip out existing gear." PO Appeal. Br. Ex. B-5, 2.

Regarding the statements made in the Short Declaration regarding certain conversations between one Sami Saydjari, alleged to be a program manager for DARPA, and Edmund Munger, a co-inventor of the '151 Patent (Short Decl. ¶ 15), we accord these statements little weight as such amount to unverified third-party conversations.

4. *Commercial Success*

Patent Owner contends the claims of the '151 Patent have experienced commercial success in view of licensing agreements with multiple companies. PO Appeal Br. 73, citing Short Decl. ¶ 12; Ex A-8 1.

Cisco contends Patent Owner has presented no evidence that the broad portfolio licenses provide evidence that the claims under reexamination were a primary motivating factor for the licensing agreements. Cisco Resp't Br. 29.

We agree with Cisco. The alleged licensing agreements apparently include multiple patents, and there is insufficient evidence that such agreements were as a result of the particular features recited in the claims of the '151 Patent. As a result, Patent Owner has provided insufficient nexus between the claims of the '151 Patent and commercial success.

5. *Praise and Acceptance by Others*

Patent Owner argues those in the industry have praised the inventions either by stating praise or investing in the technology or licensing it. PO Appeal Br. 73–74, citing Short Decl. ¶¶ 7, 12, 16.

Cisco contends Patent Owner presents no documentary evidence that the '151 Patent claims were praised or accepted by the industry, only third-hand statements.

We agree with Cisco. The statements made in the Short Declaration regarding the alleged investments made by SAIC in the technology leading to the '151 Patent as well as the statement that SAIC spent one-third of its total patent portfolio efforts on a patent portfolio including the '151 Patent (Short Decl. ¶ 7) are uncorroborated and do not particularly address any particular praise for the features in the claims of the '151 Patent. In addition, portfolio licensing agreements (Short Decl. ¶¶ 12, 16) do not demonstrate praise for the particular features recited in the claims of the '151 Patent. With respect to the alleged study done by CSMG praising the inventors and alleged praise and significant interest expressed by Jim Rutt at Network Solutions (Short Decl. ¶ 16), such statements are uncorroborated and do not relate any praise or acceptance to the particular features of the claims of the ’151 Patent.

6. Weighing the Evidence of Objective Indicia

After considering the evidence both in favor of non-obviousness and in favor of obviousness, we determine that in view of the deficiencies discussed above with respect to the evidence of non-obviousness, the evidence of obviousness carries more weight. That is, the combinations of prior art discussed above (the combinations of Aziz and Edwards; Kiuchi and Martin; Aziz, Edwards, and Martin; Kiuchi and Edwards; Kiuchi, Edwards, and Martin; and Beser and Kent) are stronger than the evidence tending to show claims of the ’151 Patent would not have been obvious for the reasons previously discussed.

IV. CISCO’S APPEAL

A. Anticipation – Wesinger (Issue 9)

The Examiner withdrew the rejection of claims 1–4, 6–10, and 12–16 as anticipated by Wesinger, finding Patent Owner’s argument persuasive that Wesinger’s disclosure of determining what security rules to apply based on a host (domain) name is not the same as “determining whether the intercepted DNS request corresponds to a secure server.” Ans. 92. The Examiner stated that a DNS entry for the “accessing machine” in Wesinger is not the same as a DNS request sent by the client as recited in the claims. *Id.* at 92–93.

Cisco contends the Examiner incorrectly withdrew the rejection of the claims because Wesinger discloses a firewall performs an allow or deny determination as part of the DNS request process. Cisco Appeal Br. 5–8, citing Wesinger Fig. 7. In particular, Cisco argues Wesinger discloses an "access rules database" that, as Cisco points out, "govern[s] access to and through the virtual host, i.e., which connections will be allowed and which connections will be denied." Cisco Appeal Br. 5, citing Wesinger col. 15, ll. 19–28 (emphasis omitted).

Patent Owner contends Wesinger's process occurs when the firewall receives a connection request and is not based on any DNS request. PO Resp't Br. 3, citing Wesinger col. 16, ll. 22–28, col. 9, ll. 16–24, and col. 15, ll. 5–19; Keromytis Decl. ¶¶ 112–113. Patent Owner contends Wesinger's firewall system makes no determination whether a connection request corresponds to a secure server, because every server in Wesinger is supported by a firewall and presumably secure, where Wesinger implements a security policy for each virtual host. *Id.* at 3–4, citing Keromytis Decl. ¶¶ 114–117. Patent Owner argues "at best, Wesinger analyzes whether the remote host (i.e., client) requesting a connection is secure . . . [b]ut [Wesinger] does not additionally disclose determining whether a connection request, much less a DNS request, corresponds to a secure server." *Id.* at 4, citing Keromytis Decl. ¶ 117 (emphases omitted).

We agree with the Examiner and Patent Owner that Wesinger fails to disclose analyzing the remote host (host requesting connection or client) in order to determine whether it is secure and that Wesinger fails to disclose determining whether a DNS request corresponds to a secure server. In particular, Wesinger discloses in addition to checking the remote host requesting the connection (client) to determine the level of access scrutiny, checking the virtual host (destination server) to determine whether the remote host (client) is allowed access via the allow and deny databases. Wesinger, col. 16, ll. 43–60. In other words, although Wesinger determines whether the client is allowed to access a secure server, such a determination does not depend on whether the requested server is secure or nonsecure, but rather whether the client has sufficient privileges to access the requested server. Thus, Wesinger does not disclose determining whether the virtual host itself corresponds to a secure server; rather, Wesinger only discloses evaluating the security privileges pertaining to the host requesting connection (i.e., the client). In this regard, Cisco's citation to Figure 7 of Wesinger to illustrate certain allow/deny "rules" (Cisco Appeal Br. 6–8) does not speak to whether the server itself is secure, but rather only determines under what particular conditions the remote host (client) may access the virtual host (secure server).

Accordingly, we affirm the Examiner's determination that Wesinger does not anticipate the claims.

B. Obviousness – Wesinger and Edwards (Issue 16)

Cisco relies on the same arguments for the combination of Wesinger and Edwards as discussed above for the anticipation rejection based on Wesinger (Issue 9). Cisco Appeal Br. 14–16. Accordingly, for similar reasons as discussed above, we affirm the Examiner's decision to withdraw the rejection of the claims as obvious over Wesinger and Edwards.

C. Obviousness – Claims 5 and 11 Wesinger or Wesinger in view of Edwards and Martin (Issues 10 and 17)

As discussed above, claims 5 and 11 depend from claims 1 and 7, respectively. As pointed out by Patent Owner, Martin is not relied upon to remedy any deficiencies of Wesinger or Wesinger and Edwards. PO Resp't Br. 6, 13. Accordingly, we affirm the Examiner's decision not to reject claims 5 and 11 for the reasons discussed above with respect to claims 1 and 7.

D. Anticipation – Blum (Issue 11)

The Examiner found that Blum discloses determining only whether a server is local or remote, and does not disclose determining whether a server is secure, and as such Blum further fails to disclose forwarding a DNS request to a DNS function that returns an IP address of a nonsecure computer when the DNS request does not correspond to a secure server. Ans. 93–94; ACP 62–64.

Cisco contends Blum discloses intercepting a DNS request, determining whether the DNS request corresponds to a remote server, and then determining if there is a protocol filter, such as a Secure Sockets Layer (SSL) protocol associated with the connection request. Cisco Appeal Br. 9–10, citing Blum, col. 1, ll. 46–48, col. 3, ll. 42–44, col. 5, ll. 23–27, col. 6, ll. 50–51, col. 9, ll. 19–23, 33–35.

In addition to arguing Blum only discloses determining whether a server is local or remote, Patent Owner contends Blum does not address the security of the remote servers at all, and that the SSL protocol is used in client programs not in communication processes with remote servers. PO Resp't Br. 6–8, citing Keromytis Decl. ¶¶ 150–151.

Ultimately, we agree with the Examiner that Blum does not disclose the specific arrangement recited in claim 1. That is, we agree with the Examiner that Blum fails to disclose forwarding a DNS request to a DNS function that returns an IP address of a nonsecure computer when the DNS request does not correspond to a secure server as recited in the claim.

Contrary to Patent Owner's arguments and the statements made in the Keromytis Declaration, Blum does not limit the application of protocol filters to local communications. In particular, Blum discloses "the transparent proxy application 355 checks to see if there is a protocol filter 520 associated with the native protocol of the connection request ***or with a port indicated in the connection request***." Blum, col. 9, ll. 33–35 (emphasis added); Fig. 5. Blum then discloses the protocol filter is applied and if the communication is allowed to be established, the connection is established "between the requesting client application 325 ***and the remote server identified in the communication request***." *Id.* at col. 9, ll. 35–46 (emphasis added). Blum discloses well-known protocols include Secure Sockets Layer (SSL) protocols. *Id.* at col. 1, ll. 46–50.

However, Blum does not provide any particular discussion regarding the use of SSL protocols in the context of the method described therein, nor does Blum particularly disclose forwarding a DNS request to a DNS function following a determination that a DNS request does not correspond to a secure server. In an anticipation rejection, "'the [prior art] reference must clearly and unequivocally disclose the claimed [invention] or direct those skilled in the art to the [invention] without *any* need for picking, choosing, and combining various disclosures not directly related to each other by the teachings of the cited reference.'" *Net MoneyIN v. Verisign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008) (quoting *In re Arkley*, 455 F.2d 586, 587 (CCPA 1972)) (alterations in original).

As a result, we affirm the Examiner's decision to withdraw the rejection of the claims as anticipated by Blum.

E. Obviousness – Claims 2–4, 6, 8–10, 12, 14–16 Aziz and Edwards (Issue 12)

1. Claims 2–4, 8–10, and 14–16

Claim 2 is representative of the limitations argued by Cisco, and recites, in pertinent part,

> [t]he data processing device of claim 1, wherein step (iii) comprises the steps of: (a) determining whether the client is authorized to access the secure server; and when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and client.

Cisco argues the Examiner improperly withdrew the rejection of claim 2, because the Examiner overlooked Aziz discloses that when the client computer needs to communicate via an encrypted channel with the inside host 140, but does not have the necessary record for inside the host 140, the resolver makes additional inquiries. Cisco Appeal Br. 12, citing Aziz, col. 11, l. 63 – col. 12, l. 33, Fig. 4C. Cisco argues also additional queries are sent to inside name server 130, located behind a firewall, which renders obvious "sending a request to the secure server to establish an encrypted channel between the secure server and the client." *Id.*, citing Aziz, col. 12, ll. 9–11.

Patent Owner contends Aziz does not disclose sending a request to the secure server to establish an encrypted channel and Cisco's position that inside name server 130 corresponds to the claimed secure server is unsupported. PO Resp't Br. 10. Patent Owner contends Edwards also does not disclose or suggest this feature. *Id.*

Although we do not agree with Patent Owner's position that inside name server 130 does not correspond to the secure server recited in claim 2, we agree with the Examiner and Patent Owner that Aziz fails to disclose sending a request to a secure server after determining whether the client is authorized to access the secure server. Aziz discloses only that "resolver 225 makes additional queries (not shown in Fig. 4C) as necessary." Aziz, col. 12, ll. 25–27. Moreover, Aziz discloses the authorized client "will be used herein to refer to a client that is configured to use the invention and whose communications will be allowed through by the firewall for the protected hosts with which the authorized client communicates." *Id.* at col. 3, ll. 62–67. Thus, we agree with Patent Owner that Cisco's position Aziz discloses sending a request to access a secure server is not sufficiently supported.

Because claims 3, 4, 9, 10, 15, and 16 depend either directly or indirectly from claims 2, 8, and 14, respectively, we affirm the Examiner's decision to withdraw the rejection of those claims for similar reasons as discussed for claims 2, 8, and 14.

2. *Claims 6 and 12*

Claims 6 and 12 recite "the secure server avoids sending a true IP address of the secure server to the client."

Cisco contends Edwards discloses service interceptors, which provides a way to avoid sending a true address of a requested service to the client by preventing services in the internal network accidently subverting security by handing object references to a client in the outside network. Cisco Appeal Br. 13, citing Edwards 932, 936. Cisco argues because Aziz discloses an embodiment where the network topology behind firewall 110 is hidden, it would have been obvious to improve Aziz's system by using the further topology-hiding technique of Edwards. *Id.*, citing Aziz, col. 11, ll.

64–65. Thus, Cisco contends rather than returning the true IP address of a requested server, the combination of Aziz and Edwards suggest returning a false IP address corresponding to a service interceptor. *Id.*

Patent Owner contends Cisco only considers part of the claims, the portion reciting not sending a true IP address, and the combination of Aziz and Edwards does not disclose "automatically initiating the encrypted channel between the client and the secure server." PO Resp't Br. 11–12.

The Examiner determined Aziz discloses the authorized client needs the information including the hosts address. ACP 72, citing Aziz, col. 3, ll. 38–44, col. 5, ll. 57–60.

We are persuaded by Cisco's arguments. In particular, as Cisco points out, Edwards discloses a name service receives a request from the CORBAweb plugin to resolve a name, it will return a reference to a service interceptor instead of the actual service, which is consistent with the disclosure in Aziz that the network topology behind the firewall 110 is hidden. Edwards 932, 936; Aziz, col. 11, ll. 64–65. Thus, in view of our discussion above that the combination of Aziz and Edwards renders obvious claim 1, we are of the opinion that the additional disclosure in Edwards that the naming service interceptor prevents services in the internal network accidently subverting security (Edwards 936), renders obvious avoiding sending a true IP address of the secure server to the client.

Accordingly, we reverse the Examiner's decision not to reject claims 6 and 12.

F. Anticipation – Claims 6 and 12 Kiuchi (Issue 7)

Claims 6 and 12 depend from claims 1 and 7, respectively. As discussed above, we reversed the Examiner's rejection of claims 1 and 7 as anticipated by Kiuchi. Accordingly, we affirm the Examiner's decision not to reject claims 6 and 12 for similar reasons.

V. CONCLUSION

In summary, the status of the Adopted Rejections is as follows:

Claim(s) Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Affirmed**	**Reversed**
1, 2, 5, 7, 8, 11, 13, 14	102(b)	Aventail Connect v3.01	13, 14	1, 2, 5, 7, 8, 11
1, 2, 5, 7, 8, 11, 13, 14	102(b)	AutoSOCKS	13, 14	1, 2, 5, 7, 8, 11
1, 2, 5, 7, 8, 11, 13, 14	103(a)	Beser, Kent	1, 2, 5, 7, 8, 11, 13, 14	
1–4, 7–10, 13–16	102(b)	Kiuchi	13, 14	1–4, 7–10, 15, 16
5, 11	103(a)	Kiuchi, Martin		5, 11
1, 7, 13	103(a)	Aziz, Edwards	1, 7, 13	
5, 11	103(a)	Aziz, Edwards, Martin	5, 11	
1–4, 6–10, 12–16	103(a)	Kiuchi, Edwards	13, 14	1–4, 6–10, 12, 15, 16
5, 11	103(a)	Kiuchi, Edwards, Martin		5, 11
Overall Outcome Adopted Rejections			**1, 2, 5, 7, 8, 11, 13, 14**	**3, 4, 6, 9, 10, 12, 15, 16**

The status of the Non-Adopted Rejections is as follows:

Claim(s) not Rejected	35 U.S.C. §	Reference(s) /Basis	Affirmed	Reversed	New Ground Of Rejection
6, 12	102(b)	Kiuchi	6, 12		
1–4, 6–10, 12–16	102(e)	Wesinger	1–4, 6–10, 12–16		
5, 11	103(a)	Wesinger, Martin	5, 11		
1, 7, 13	102(e)	Blum	1, 7, 13		
2–4, 6, 8–10, 12, 14–16	103(a)	Aziz, Edwards	2–4, 8–10, 14–16	6, 12	6, 12
1–4, 6–10, 12–16	103(a)	Wesinger, Edwards	1–4, 6–10, 12–16		
5, 11	103(a)	Wesinger, Edwards, Martin	5, 11		
Overall Outcome Non-Adopted Rejections			**1–5, 7–11, 13–16**	**6, 12**	**6, 12**

NEW GROUND OF REJECTION

This decision contains new grounds of rejection pursuant to 37 C.F.R. § 41.77(b) which provides that "[a]ny decision which includes a new ground of rejection pursuant to this paragraph shall not be considered final for judicial review." Correspondingly, no portion of the decision is final for purposes of judicial review. A requester may also request rehearing under 37 C.F.R. § 41.79, if appropriate; however, the Board may elect to defer issuing any decision on such request for rehearing until such time that a final decision on appeal has been issued by the Board.

For further guidance on new grounds of rejection, *see* 37 C.F.R. § 41.77(b)–(g). The decision may become final after it has returned to the Board. 37 C.F.R. § 41.77(f).

37 C.F.R. § 41.77(b) also provides that the Patent Owner, WITHIN ONE MONTH FROM THE DATE OF THE DECISION, must exercise one of the following two options with respect to the new grounds of rejection to avoid termination of the appeal as to the rejected claims:

(1) *Reopen prosecution.* The owner may file a response requesting reopening of prosecution before the examiner. Such a response must be either an amendment of the claims so rejected or new evidence relating to the claims so rejected, or both.

(2) *Request rehearing*. The owner may request that the proceeding be reheard under § 41.79 by the Board upon the same record. . . .

Any request to reopen prosecution before the examiner under 37 C.F.R. § 41.77(b)(1) shall be limited in scope to the "claims so rejected." Accordingly, a request to reopen prosecution is limited to issues raised by the new ground(s) of rejection entered by the Board. A request to reopen prosecution that includes issues other than those raised by the new ground(s) is unlikely to be granted. Furthermore, should the patent owner seek to substitute claims, there is a presumption that only one substitute claim would be needed to replace a cancelled claim.

A requester may file comments in reply to a patent owner response. 37 C.F.R. § 41.77(c). Requester comments under 37 C.F.R. § 41.77(c) shall be limited in scope to the issues raised by the Board's opinion reflecting its decision to reject the claims and the patent owner's response under paragraph 37 C.F.R. § 41.77(b)(1). A newly proposed rejection is not permitted as a matter of right. A newly proposed rejection may be appropriate if it is presented to address an amendment and/or new evidence properly submitted by the patent owner, and is presented with a brief explanation as to why the newly proposed rejection is now necessary and why it could not have been presented earlier.

Compliance with the page limits pursuant to 37 C.F.R. § 1.943(b), for all patent owner responses and requester comments, is required.
The examiner, after the Board's entry of a patent owner response and requester comments, will issue a determination under 37 C.F.R. § 41.77(d) as to whether the Board's rejection is maintained or has been overcome. The proceeding will then be returned to the Board together with any comments and reply submitted by the owner and/or requester under 37 C.F.R. § 41.77(e) for reconsideration and issuance of a new decision by the Board as provided by 37 C.F.R. § 41.77(f).

AFFIRMED-IN-PART; 37 C.F.R. § 41.77(b)

ELD

PATENT OWNER:

PAUL HASTINGS LLP
875 15th Street, NW
Washington, DC 20005

THIRD PARTY REQUESTER:

SIDLEY AUSTIN LLP
2021 McKinney Avenue, Suite 2000
Dallas, TX 75201

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/001,714	08/16/2011	7,490,151 B2	43614.99	3428

7590 09/28/2020

PAUL HASTINGS LLP
2050 M STREET, NW
WASHINGTON, DC 20036

EXAMINER	
YIGDALL, MICHAEL J	
ART UNIT	PAPER NUMBER
3992	
MAIL DATE	DELIVERY MODE
09/28/2020	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Requester

v.

VIRNETX, INC.,
Patent Owner

Appeal 2020-000639
Inter Partes Reexamination Control 95/001,714
Patent 7,490,151 B2
Technology Center 3900

RAE LYNN P. GUEST, *Administrative Patent Judge*.

DECISION ON PETITION

This is a decision dismissing "PATENT OWNER'S PETITION TO VACATE JUNE 23, 2020, BOARD DECISION," filed June 29, 2020 ("Petition").[1] The petition requests the vacatur of the Board's Decision of June 23, 2020. Petition 1. The petition fee of $1,940 in accordance with 37 C.F.R. § 1.20(c)(6) was charged to Patent Owner's deposit account on June 29, 2020.

FINDINGS

1. On February 24, 2016, a Right of Appeal Notice ("RAN") was entered in the merged *inter partes* reexamination proceedings 95/001,697 & 95/001,714, which rejected claims 1-16.
2. Patent Owner filed a Notice of Appeal on March 24, 2016.
3. Third Party Requester "Cisco Systems," filed a Notice of Cross-appeal on April 4, 2016, and an Appellant's Brief on June 3, 2016.
4. Patent Owner filed an Appellant's Brief on June 6, 2016.
5. Respondent briefing was completed in the merged proceeding and an Examiner's Answer was mailed on March 30, 2017.
6. Rebuttal briefing was completed on May 1, 2017.
7. On June 2, 2017, Patent Owner filed a petition to sever the merger and terminate the '1697 reexamination proceeding because the unsuccessful challenge to the '151 patent in district court by Apple Inc., the Third Party Requester in the '1697 reexamination, barred the

[1] The Petition requested relief under either 37 C.F.R. § 1.181 or under 37 C.F.R. § 41.3, as applicable. Because the relief requested is vacatur of a Board decision, the Board takes jurisdiction to decide this petition under 37 C.F.R. § 41.3.

Requester from asserting invalidity in an *inter partes* reexamination under pre-AIA 35 U.S.C. § 317(b).

8. In a Decision issued August 1, 2019, the Court of Appeals for the Federal Circuit held that Apple Inc., the Third Party Requester in related *inter partes* reexamination proceedings 95/001,788 and 95/001,789, as well as the '1697 reexamination, was estopped from filing a request for reexamination with respect to certain claims adjudicated in the same district court proceeding under 35 U.S.C. § 317(b). *VirnetX, Inc. v. Apple, Inc.*, 931 F.3d 1363 (Fed. Cir. 2019).
9. On October 16, 2019, a Decision granting Patent Owner's petition of June 2, 2017, severing the merged *inter partes* reexamination proceedings 95/001,697 & 95/001,714 and terminating reexamination with respect to claims 1-6 and 13-16 of the '151 patent in the '1697 proceeding, was mailed.
10. On November 6, 2019, an Appeal Docketing Notice was mailed in the 95/001,714 proceeding ("'1714 proceeding").
11. On November 12, 2019, Patent Owner filed a petition requesting vacatur of the February 24, 2016, Right of Appeal Notice (RAN) and issuance of a new RAN.
12. On June 23, 2020, the Patent Trial and Appeal Board ("Board") entered a Decision on appeal in the '1714 reexamination proceeding, that included new grounds of rejection under 37 C.F.R. § 41.77(b). Decision 3, 6, 58-59, 62-64.
13. The Decision on appeal, citing Section 41.77(b), notes that Appellant has one month from the date of the Decision to file a response

requesting reopening of prosecution before the Examiner or a request for rehearing by the Board. *Id.* at 63.

14. On June 29, 2020, Patent Owner filed a petition requesting vacatur of the Board's June 23, 2020 Decision, as well as a petition requesting an extension of time to respond to the Board's Decision.
15. Patent Owner filed a timely Request to Reopen Prosecution in accordance with 37 C.F.R. § 41.77(b)(1), accompanied by new evidence on July 23, 2020.
16. On August 17, 2020, a decision dismissing the petition for an extension of time was mailed as moot.
17. Requester filed Comments under § 41.77(c) on Patent Owner's request to reopen prosecution on August 21, 2020.

RELEVANT AUTHORITY

35 U.S.C. § 314(c) (pre-AIA) provides:

> SPECIAL DISPATCH.— Unless otherwise provided by the Director for good cause, all inter partes reexamination proceedings under this section, including any appeal to the Board of Patent Appeals and Interferences,[2] shall be conducted with special dispatch within the Office.

37 C.F.R. § 41.77(b),(c) provide:

> (b) Should the Board reverse the examiner's determination not to make a rejection proposed by a requester, the Board shall set forth in the opinion in support of its decision a new ground of rejection; or should the Board have knowledge of any grounds not raised in the appeal for rejecting any pending claim, it may include in its opinion a statement to that effect with its reasons for so holding, which statement shall

[2] Now known as the Patent Trial and Appeal Board.

> constitute a new ground of rejection of the claim. Any decision which includes a new ground of rejection pursuant to this paragraph shall not be considered final for judicial review. When the Board makes a new ground of rejection, the owner, within one month from the date of the decision, must exercise one of the following two options with respect to the new ground of rejection to avoid termination of the appeal proceeding as to the rejected claim:
>
> > (1) *Reopen prosecution.* The owner may file a response requesting reopening of prosecution before the examiner. Such a response must be either an amendment of the claims so rejected or new evidence relating to the claims so rejected, or both.
> >
> > (2) *Request rehearing.* The owner may request that the proceeding be reheard under § 41.79 by the Board upon the same record. The request for rehearing must address any new ground of rejection and state with particularity the points believed to have been misapprehended or overlooked in entering the new ground of rejection and also state all other grounds upon which rehearing is sought.
>
> (c) Where the owner has filed a response requesting reopening of prosecution under paragraph (b)(1) of this section, any requester, within one month of the date of service of the owner's response, may once file comments on the response. Such written comments must be limited to the issues raised by the Board's opinion reflecting its decision and the owner's response. Any requester that had not previously filed an appeal or cross appeal and is seeking under this subsection to file comments or a reply to the comments is subject to the appeal and brief fees under § 41.20(b)(1) and (2), respectively, which must accompany the comments or reply.

37 C.F.R. § 41.3 provides:

> (a) Deciding official. Petitions must be addressed to the Chief Administrative Patent Judge. A panel or an administrative patent judge may certify a question of policy to the Chief Administrative Patent Judge for decision. The Chief

Administrative Patent Judge may delegate authority to decide petitions.

(b) Scope. This section covers petitions on matters pending before the Board (§§ 41.35, 41.64, 41.103, and 41.205); otherwise, see §§ 1.181 to 1.183 of this title. The following matters are not subject to petition:

(1) Issues committed by statute to a panel, and

(2) In pending contested cases, procedural issues. See § 41.121(a)(3) and § 41.125(c).

(c) Petition fee. The fee set in § 41.20(a) must accompany any petition under this section except no fee is required for a petition under this section seeking supervisory review.

(d) Effect on proceeding. The filing of a petition does not stay the time for any other action in a Board proceeding.

(e) Time for action.

(1) Except as otherwise provided in this part or as the Board may authorize in writing, a party may:

(i) File the petition within 14 days from the date of the action from which the party is requesting relief, and

(ii) File any request for reconsideration of a petition decision within 14 days of the decision on petition or such other time as the Board may set.

(2) A party may not file an opposition or a reply to a petition without Board authorization.

37 C.F.R. § 41.35(d) provides:

(d) Documents filed during Board's jurisdiction. Except for petitions authorized by this part, consideration of any information disclosure statement or petition filed while the Board possesses jurisdiction over the proceeding will be held in abeyance until the Board's jurisdiction ends.

MPEP § 2681, Part IV provides:

Because review of the decisions of the Board relating to patentability is within the exclusive jurisdiction of the U.S. Court of Appeals for the Federal Circuit, the Board's decisions are properly reviewable on

> petition only for procedural matters and only to the extent of determining whether they involve a convincing showing of error, abuse of discretion, or policy issue appropriate for higher level determination. Reasonable rulings made by the Board on procedural matters resting in its discretion will not be disturbed upon petition. A party in disagreement with a decision of the Board on substantive merits should consider the appropriateness of filing a request for rehearing under 37 CFR 41.79 or an appeal to the U.S. Court of Appeals for the Federal Circuit.

ANALYSIS

Patent Owner's petition requesting vacatur of the Board's Decision on appeal mailed June 23, 2020, has been considered. In the petition, Patent Owner asserts that the Board Decision violates pre-AIA 35 U.S.C. § 317(b) and the Office's prior petition decision dated October 16, 2019 ("Pet. Dec."). Patent Owner contends that the petition decision provided that "[a]ny rejection which is presently applied against claims 1-6 and 13-16 of the '151 patent, i.e., the patent under reexamination, in the '1697 reexamination proceeding will not be further maintained by the Office." Pet. Dec. at 2 (emphasis in original). Patent Owner argues that in light of the petition decision, it requests vacatur of the Right of Appeal Notice ("RAN")—which included the precluded rejections—and for issuance of a new RAN. Patent Owner asserts that the Board Decision overlooked Patent Owner's November 12, 2019, petition seeking vacatur of the RAN filed after severing of the merged reexaminations in accordance with the Petition Decision. Patent Owner argues that the Board Decision adopted rejections that were applied against claims 1–6 and 13–16 of the '151 patent in the '1697 reexamination proceeding. Patent Owner asserts that the Board

Decision exceeded the Board's statutory authority in contravention of the Petition Decision and in violation of pre-AIA 35 U.S.C. § 317(b).

A petition requesting review of a Board Decision is appropriate only for procedural matters, and only to the extent of determining whether it involves a convincing showing of error, an abuse of discretion, or a policy issue appropriate for higher level determination. Reasonable rulings made by the Board on procedural matters resting in its discretion will not be disturbed upon petition. *See* MPEP § 2681, Part IV.

The Board correctly did not address the November 17, 2019, petition seeking vacatur of the RAN.[3] Pursuant to 37 C.F.R. § 41.35(d), except for petitions authorized by part 41, a petition filed while the Board has jurisdiction will be held in abeyance until the Board's jurisdiction ends. Because the petition was filed after the Board entered an Appeal Docketing Notice on November 6, 2019, the petition is held in abeyance.[4]

The petition decision of October 16, 2019, which granted severance of the merged reexaminations states in pertinent part:

> Patent owner's June 2, 2017 renewed petition under 37 CFR 1.182 to sever the merger of 95/001,697 and 95/001,714 (the merged proceeding), **and terminate reexamination of claims 1-6 and 13-16** of U.S. Patent No. 7,490,151 (the '151 patent) **in *inter partes***

[3] The Petition Decision severing the merged reexaminations indicated that a new RAN would be issued in the '1697 reexamination only. New RANs have been issued in the '1697 reexamination on November 6, 2019 and June 15, 2020. However, in the '1714 reexamination, the Petition Decision severing the merged reexaminations did not order a new RAN to be issued.

[4] We note that the November 17, 2019, petition seeking vacatur of the RAN was silent as to the regulatory provision under which it was filed. Because the Board does not issue RANs, it would be unreasonable to find that the petition sought relief under 37 C.F.R. § 41.3.

> **reexamination proceeding control number 95/001,697** (the '1697 proceeding) is granted.
>
> The merged proceeding of *inter partes* reexamination control numbers 95/001,697 and 95/001,714 is hereby severed.
>
> Prosecution of claims 1-16 of the '151 patent in *inter partes* reexamination control number 95/001,714 will continue.
>
> **The estoppel provisions of pre-AIA 35 U.S.C. 317(b) apply to any rejection of claims 1-6 and 13-16 of the '151 patent in the '1697 proceeding.** Any rejection which is presently applied against claims 1-6 and 13-16 of the '151 patent, i.e., the patent under reexamination, in the '1697 reexamination proceeding will not be further maintained by the Office. **No further rejection of claims 1-6 and 13-16 of the '151 patent will be made in the '1697 reexamination proceeding.**
>
> **The determination in this decision not to maintain, in the '1697 reexamination proceeding, any rejection of claims 1-6 and 13-16 of the '151 patent is made pursuant to the estoppel provisions of pre-AIA 35 U.S.C. 317(b)**, and is not a "non-adoption of" or a "determination not to make" such a rejection within the meaning of 37 CFR 41.61. **For this reason, any notice of appeal or cross-appeal of the present determination in the '1697 reexamination proceeding not to make or maintain a rejection of claims 1-6 and 13-16 of the '151 patent will be held to be defective.**
>
> Prosecution in the '1697 reexamination proceeding with respect to claims 7-12 of the '151 patent will continue. The '1697 reexamination proceeding will be forwarded to the Central Reexamination Unit (CRU) for processing and issuance of a new Right of Appeal Notice (RAN) consistent with this decision.

Renewed Pet. Dec. 2 (emphasis added).

In its petition, Patent Owner appears to be suggesting that the estoppel not only applies to Apple in the '1697 proceeding, but also extends to Cisco in the '1714 proceeding. However, the Federal Circuit rejected this notion in *In re: Affinity Labs* 856 F.3d 883, 893 (Fed. Cir. 2017):

> We also find no basis in the statute for Affinity's argument that the final decision in the Volkswagen litigation should have preclusive

> effect on the reexaminations requested by King or Apple. By its plain and unambiguous terms, pre-AIA section 317(b) . . . **imposes the aforementioned limitations only on a requester that was a party to the civil action or its privies. Because Apple was neither a party to the Volkswagen litigation nor was there any evidence Apple was Volkswagen's privy, we also find no error in the PTO's decision not to terminate the *inter partes* reexamination requested by Apple.**
>
> Affinity's invocation of the general policy goal of preventing abusive reexamination practices cannot override the statute. . . . The plain language of the statute does not permit extending the reach of estoppel as far as Affinity suggests. Moreover, we have reviewed the legislative history and find no such extraordinary showing sufficient to overcome the statute's plain language.

(Emphasis added) (internal citations omitted).

Patent Owner further contends that:

> the Petition Decision held that "[a]ny rejection which is presently applied against claims 1-6 and 13-16 of the '151 patent, i.e., the patent under reexamination, in the '1697 reexamination proceeding **will not be further maintained by the Office**." (Petition Decision at 2 (emphasis in original).) In other words, the rejections that Apple had proposed against claims 1-6 and 13-16 (based on Aventail Connect v3.01, Aventail AutoSOCKS, and Beser/Kent) could no longer be maintained. Thus, the Board did not have authority to consider, much less affirm, the rejections proposed by Apple in the '1697 reexamination proceeding against the same claims challenged in the '1714 reexamination proceeding (where Cisco had not proposed those rejections).

Pet. 5–6.

The appeal of claims 1–16 in the '1714 proceeding is based on the rejections of claims 1–16 (and non-adoption of proposed rejections), set forth by the Examiner in the RAN of February 24, 2016. Once the Office adopts proposed rejections, the rejections are the Examiner's rejections, i.e., the position of the Office. Indeed, in rejecting claims in *inter partes* reexamination, the Examiner is not limited only to the prior art or rationales proposed by a requesting party, in this case Cisco. *See* 37 C.F.R. §§ 1.937 ("The *inter partes* reexamination proceeding will be conducted in accordance with §§ 1.104 through 1.116. . . ."), 1.104 ("In rejecting claims for want of novelty or for obviousness, the examiner must cite the best references at his or her command."); MPEP § 2656 (While the request for *inter partes* reexamination will be "the primary source of prior art[,] . . . the examiner must also consider patents and printed publications: (A) cited by another reexamination requester under 37 C.F.R. § 1.510 or 37 C.F.R. § 1.915; (B) cited by the patent owner under a duty of disclosure (37 CFR 1.933) in compliance with 37 CFR 1.98; (C) discovered by the examiner in searching; (D) of record in the patent file from earlier examination; . . .").

Although the '1697 proceeding was severed from the instant appeal subsequent to the RAN, the Petition Decision ordered that "[p]rosecution of claims 1–16 of the '151 patent in *inter partes* reexamination control number 95/001,714 will continue." Renewed Pet. Dec. 2, 14, 15. The Petition Decision only ordered a new RAN be mailed in the '1697 reexamination proceeding, not in the '1714 reexamination proceeding. *Id.* Therefore, Patent Owner's assertion that the Board exceeded its statutory authority because the Board's Decision in the '1714 reexamination decided the merits of rejections, which were not permitted to be applied against claims 1–6 and 13–16 of the '151 patent in the '1697 reexamination proceeding, is not consistent with the Petition Decision. The Petition Decision only explains how the estoppel provision will be applied to rejections in the issuance of a new RAN in the '1697 reexamination proceeding.

Nor is it inconsistent with 35 U.S.C. § 317(b) for the Examiner to maintain rejections over the best references at his or her command. 35 U.S.C. § 317(b) only requires actions consistent with the Petition Decision severing the merged reexaminations, and does not require withdrawal of any specific rejections maintained by the Examiner in the '1714 reexamination filed by Cisco.

A review of the Board's Decision for procedural matters, only to the extent of determining whether it involves a convincing showing of error, an abuse of discretion, or a policy issue reveals no error. The Board's Decision on appeal of June 23, 2020, stands.

The petition is **dismissed**.

DECISION

In view of the foregoing, Patent Owner's petition requesting vacatur of the Board's Decision is DISMISSED.

Patent Owner's request to reopen prosecution under 37 C.F.R. § 41.77(b)(1) of July 23, 2020, and Requester's Comments under 37 C.F.R. § 41.77(c) of August 21, 2020, are before the Board and will be considered in due course.

PATENT OWNER:

PAUL HASTINGS LLP
2050 M STREET, NW
WASHINGTON, DC 20036

THIRD-PARTY REQUESTER:

SIDLEY AUSTIN, LLP
2021 MCKINNEY AVENUE, SUITE 2000
DALLAS, TX 75201

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Petitioner,

v.

VIRNETX, INC.,
Patent Owner.

Appeal 2020-000639
Inter Partes Reexamination Control 95/001,714
Patent 7,490,151 B2
Technology Center 3900

Before ANDREI IANCU, *Under Secretary of Commerce for Intellectual Property and Director of the United States Patent and Trademark Office*, ANDREW HIRSHFELD, *Commissioner for Patents*, and SCOTT R. BOALICK, *Chief Administrative Patent Judge.*

PER CURIAM.

ORDER

The Office has received a request for Precedential Opinion Panel (POP) review of an issue raised in this case. The request was referred to the POP panel referenced above.

Upon consideration of the request, it is ORDERED that:

The request for POP review is denied; and

FURTHER ORDERED that the original panel maintains authority over all matters, including considering the submitted rehearing request.

PATENT OWNER:

PAUL HASTINGS LLP
2050 M STREET, NW
WASHINGTON, DC
20036

THIRD-PARTY REQUESTER:

SIDLEY AUSTIN, LLP
2021 MCKINNEY AVENUE, SUITE 2000
DALLAS, TX 75201

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/001,714	08/16/2011	7,490,151 B2	43614.99	3428

7590 04/27/2021

PAUL HASTINGS LLP
2050 M STREET, NW
WASHINGTON, DC 20036

EXAMINER	
RALIS, STEPHEN J	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
04/27/2021	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/001,714	08/16/2011	7,490,151 B2	43614.99	3428

7590 04/27/2021

SIDLEY AUSTIN, LLP
2021 MCKINNEY
AVENUE, SUITE 2000
DALLAS, TX 75201

EXAMINER	
RALIS, STEPHEN J	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
04/27/2021	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Requester

v.

VIRNETX, INC.,
Patent Owner

Appeal 2020-000639
Inter Partes Reexamination Control 95/001,714
Patent 7,490,151 B2
Technology Center 3900

DECISION ON PETITION

This is a decision denying "PATENT OWNER'S REQUEST FOR REHEARING, OR IN THE ALTERNATIVE, REQUEST FOR REVIEW OF DECISION DISMISSING PETITION TO VACATE JUNE 23, 2020 BOARD DECISION," filed October 13, 2020 ("petition"). The petition requests rehearing of the Board's decision of September 28, 2020, which dismissed the petition of June 29, 2020, by either a POP panel review under SOP2, or in the alternative, by supervisory review under 37 C.F.R. § 41.3 by the Chief Judge. Petition 1-2. No petition fee is due for a petition requesting supervisory review in accordance with 37 C.F.R. § 41.3(c).

PROCEDURAL HISTORY

1. On February 24, 2016, a Right of Appeal Notice ("RAN") was mailed in the merged *inter partes* reexamination proceedings 95/001,697 & 95/001,714, which rejected claims 1–16.
2. Patent Owner filed a Notice of Appeal on March 24, 2016.
3. Third Party Requester "Cisco Systems," filed a Notice of Cross-appeal on April 4, 2016, and an Appellant's Brief on June 3, 2016.
4. Patent Owner filed an Appellant's Brief on June 6, 2016.
5. Respondent briefing was completed in the merged proceeding and an Examiner's Answer was mailed on March 30, 2017.
6. Rebuttal briefing was completed on May 1, 2017.
7. On June 2, 2017, Patent Owner filed a petition to sever the merger and terminate the '1697 reexamination proceeding because in the unsuccessful challenge to the '151 patent in district court by Apple Inc., the Third Party Requester in the '1697 reexamination, barred the Requester from asserting invalidity in an *inter partes* reexamination under pre-AIA 35 U.S.C. § 317(b).
8. In a Decision issued August 1, 2019, the Court of Appeals for the Federal Circuit held that Apple Inc., the Third Party Requester in related *inter partes* reexamination proceedings 95/001,788 and 95/001,789, as well as the '1697 reexamination, was estopped from filing a request for reexamination with respect to certain claims

adjudicated in the same district court proceeding under 35 U.S.C. § 317(b). *VirnetX, Inc. v. Apple, Inc.*, 931 F.3d 1363 (Fed. Cir. 2019).

9. On October 16, 2019, a Decision granting Patent Owner's petition of June 2, 2017, was mailed, which severed the merged *inter partes* reexamination proceedings 95/001,697 & 95/001,714 and terminated reexamination with respect to claims 1–6 and 13–16 of the '151 patent in the '1697 proceeding requested by Apple.
10. On November 6, 2019, an Appeal Docketing Notice was mailed in the 95/001,714 proceeding ("'1714 proceeding").
11. On November 12, 2019, Patent Owner filed a petition requesting vacatur of the February 24, 2016, Right of Appeal Notice (RAN) and issuance of a new RAN[1].
12. On June 23, 2020, the Patent Trial and Appeal Board ("Board") entered a Decision on appeal in the '1714 reexamination proceeding, that included new grounds of rejection under 37 C.F.R. § 41.77(b). Decision 3, 6, 58–59, 62–64.
13. The Decision on appeal, citing Section 41.77(b), notes that Appellant has one month from the date of the Decision to file a response requesting reopening of prosecution before the Examiner or a request for rehearing by the Board. *Id.* at 63.

[1] Pursuant to 37 C.F.R. § 41.35(d), except for petitions authorized by part 41, a petition filed while the Board has jurisdiction will be held in abeyance until the Board's jurisdiction ends. Because the petition was filed after the Board entered an Appeal Docketing Notice on November 6, 2019, the petition is held in abeyance.

14. On June 29, 2020, Patent Owner filed a petition requesting vacatur of the Board's June 23, 2020 Decision, as well as a petition requesting an extension of time to respond to the Board's Decision.
15. Patent Owner filed a timely Request to Reopen Prosecution in accordance with 37 C.F.R. § 41.77(b)(1), accompanied by new evidence on July 23, 2020.
16. On August 17, 2020, a decision dismissing the petition for an extension of time was mailed.
17. Requester filed Comments under § 47.77(c) on Patent Owner's request to reopen prosecution on August 21, 2020.
18. On September 28, 2020, a decision dismissing the petition requesting vacatur was mailed.
19. Patent Owner filed the instant petition on October 13, 2020.
20. Patent Owner filed a POP request on December 1, 2020.
21. On December 8, 2020, a decision denying the POP request was mailed.

RELEVANT AUTHORITY

35 U.S.C. § 314(c) (pre-AIA) provides:

> SPECIAL DISPATCH.— Unless otherwise provided by the Director for good cause, all *inter partes* reexamination proceedings under this section, including any appeal to the

> Board of Patent Appeals and Interferences,[2] shall be conducted with special dispatch within the Office.

37 C.F.R. § 41.3 provides:

> (a) Deciding official. Petitions must be addressed to the Chief Administrative Patent Judge. A panel or an administrative patent judge may certify a question of policy to the Chief Administrative Patent Judge for decision. The Chief Administrative Patent Judge may delegate authority to decide petitions.
> (b) Scope. This section covers petitions on matters pending before the Board (§§ 41.35, 41.64, 41.103, and 41.205); otherwise, see §§ 1.181 to 1.183 of this title. The following matters are not subject to petition:
>> (1) Issues committed by statute to a panel, and
>> (2) In pending contested cases, procedural issues. *See* § 41.121(a)(3) and § 41.125(c).
>
> (c) Petition fee. The fee set in § 41.20(a) must accompany any petition under this section except no fee is required for a petition under this section seeking supervisory review.
> (d) Effect on proceeding. The filing of a petition does not stay the time for any other action in a Board proceeding.
> (e) Time for action.
>> (1) Except as otherwise provided in this part or as the Board may authorize in writing, a party may:
>>> (i) File the petition within 14 days from the date of the action from which the party is requesting relief, and
>>> (ii) File any request for reconsideration of a petition decision within 14 days of the decision on petition or such other time as the Board may set.

[2] Now known as the Patent Trial and Appeal Board.

> (2) A party may not file an opposition or a reply to a petition without Board authorization.

37 C.F.R. § 41.77(b),(c) provides:

> (b) Should the Board reverse the examiner's determination not to make a rejection proposed by a requester, the Board shall set forth in the opinion in support of its decision a new ground of rejection; or should the Board have knowledge of any grounds not raised in the appeal for rejecting any pending claim, it may include in its opinion a statement to that effect with its reasons for so holding, which statement shall constitute a new ground of rejection of the claim. Any decision which includes a new ground of rejection pursuant to this paragraph shall not be considered final for judicial review. When the Board makes a new ground of rejection, the owner, within one month from the date of the decision, must exercise one of the following two options with respect to the new ground of rejection to avoid termination of the appeal proceeding as to the rejected claim:
>
> > (1) *Reopen prosecution.* The owner may file a response requesting reopening of prosecution before the examiner. Such a response must be either an amendment of the claims so rejected or new evidence relating to the claims so rejected, or both.
> >
> > (2) *Request rehearing.* The owner may request that the proceeding be reheard under § 41.79 by the Board upon the same record. The request for rehearing must address any new ground of rejection and state with particularity the points believed to have been misapprehended or overlooked in entering the new ground of rejection and also state all other grounds upon which rehearing is sought.
>
> (c) Where the owner has filed a response requesting reopening of prosecution under paragraph (b)(1) of this section, any requester, within one month of the date of service of the owner's response, may once file comments on the response. Such written comments must be limited to the issues raised by the Board's opinion reflecting its decision and the owner's

> response. Any requester that had not previously filed an appeal or cross appeal and is seeking under this subsection to file comments or a reply to the comments is subject to the appeal and brief fees under § 41.20(b)(1) and (2), respectively, which must accompany the comments or reply.

37 C.F.R. § 41.79 provides:

> (a) Parties to the appeal may file a request for rehearing of the decision within one month of the date of:
>
> (1) The original decision of the Board under § 41.77(a),
> (2) The original § 41.77(b) decision under the provisions of § 41.77(b)(2),
> (3) The expiration of the time for the owner to take action under § 41.77(b)(2), or
> (4) The new decision of the Board under § 41.77(f).
>
> (b)
>
> (1) The request for rehearing must state with particularity the points believed to have been misapprehended or overlooked in rendering the Board's opinion reflecting its decision. Arguments not raised in the briefs before the Board and evidence not previously relied upon in the briefs are not permitted in the request for rehearing except as permitted by paragraphs (b)(2) and (b)(3) of this section.
> (2) Upon a showing of good cause, appellant and/or respondent may present a new argument based upon a recent relevant decision of either the Board or a Federal Court.
> (3) New arguments responding to a new ground of rejection made pursuant to § 41.77(b) are permitted.
>
> (c) Within one month of the date of service of any request for rehearing under paragraph (a) of this section, or any further request for rehearing under paragraph (d) of this section, the owner and all requesters may once file comments in opposition to the request for rehearing or the further request for rehearing. The comments in opposition must be limited to the issues raised in the request for rehearing or the further request for rehearing.
>
> (d) If a party to an appeal files a request for rehearing under paragraph (a) of this section, or a further request for rehearing under this section, the Board shall render a decision on the request for rehearing. The

> decision on the request for rehearing is deemed to incorporate the earlier opinion reflecting its decision for appeal, except for those portions specifically withdrawn on rehearing and is final for the purpose of judicial review, except when noted otherwise in the decision on rehearing. If the Board opinion reflecting its decision on rehearing becomes, in effect, a new decision, and the Board so indicates, then any party to the appeal may, within one month of the new decision, file a further request for rehearing of the new decision under this subsection. Such further request for rehearing must comply with paragraph (b) of this section.
> (e) The times for requesting rehearing under paragraph (a) of this section, for requesting further rehearing under paragraph (c) of this section, and for submitting comments under paragraph (b) of this section may not be extended.

MPEP § 2681, Part IV provides:

> Because review of the decisions of the Board relating to patentability is within the exclusive jurisdiction of the U.S. Court of Appeals for the Federal Circuit, the Board's decisions are properly reviewable on petition only for procedural matters and only to the extent of determining whether they involve a convincing showing of error, abuse of discretion, or policy issue appropriate for higher level determination. Reasonable rulings made by the Board on procedural matters resting in its discretion will not be disturbed upon petition. A party in disagreement with a decision of the Board on substantive merits should consider the appropriateness of filing a request for rehearing under 37 CFR 41.79 or an appeal to the U.S. Court of Appeals for the Federal Circuit.

MPEP § 2686.04, Part IV provides in pertinent part:

> The issuance of a final court decision **upholding validity** during an *inter partes* reexamination, where the person who filed the request **was not a party to the litigation**, will have no binding effect on the examination of the reexamination. This is because the court stated in *Ethicon v. Quigg,* 849 F.2d 1422, 1428, 7 USPQ2d 1152, 1157 (Fed.

> Cir. 1988) that the Office is not bound by a court's decision upholding validity and should continue the reexamination. The court noted that district courts and the Office use different standards of proof in determining invalidity and unpatentability, and thus, on the same evidence, could quite correctly come to different conclusions. Specifically, invalidity in a district court must be shown by "clear and convincing" evidence, whereas in the Office it is sufficient to show non-patentability by a "preponderance" of the evidence. Since the "clear and convincing" standard is harder to satisfy than the "preponderance standard," a court's decision upholding validity is not controlling. Deference will, however, ordinarily be accorded to the factual findings of the court, where the evidence before the Office and the court is the same. If sufficient reasons are present, claims upheld by the court may be rejected in reexamination.

ANALYSIS

Patent Owner's petition requests rehearing of the Board's decision of September 28, 2020, which dismissed the petition of June 29, 2020, by either a POP panel review under SOP2, or in the alternative, by supervisory review under 37 C.F.R. § 41.3 by the Chief Judge. Petition 1-2. Patent Owner indicates that "[i]f it is determined that POP review is not appropriate, VirnetX requests supervisory review under 37 C.F.R. § 41.3 by the Chief Judge and requests that the Chief Judge consider convening the POP to decide the issues presented in this request." *Id.* at 2. However, POP review was denied in an Order mailed on December 8, 2020, and the Chief Judge has delegated the rehearing of the petition decision.

Patent Owner's request for reconsideration of the Board's decision of September 28, 2020, which dismissed the petition of June 29, 2020, has been considered fully. Petition 1. A petition requesting review of a Board decision is appropriate only for procedural matters, and **only to the extent of** determining whether it involves a convincing showing of error, an abuse of discretion, or a policy issue appropriate for higher level determination. Reasonable rulings made by the Board on procedural matters resting in its discretion will not be disturbed upon petition. *See* MPEP § 2681, Part IV.

In the petition, Patent Owner asserts that the Board's petition decision contravenes pre-AIA 35 U.S.C. § 317(b) and the Federal Circuit's construction of that statutory provision in *Fairchild (Taiwan) Corp. v. Power Integrations, Inc.*, 854 F.3d 1364 (Fed. Cir. 2017), and *VirnetX Inc. v. Apple Inc.*, 931 F.3d 1363 (Fed. Cir. 2019). Petition 1. Patent Owner contends that the Board's petition decision end-runs Section 317(b), in defiance of the Federal Circuit's binding guidance, and requests that the decision be reversed. *Id.*

Patent Owner argues that:

> The Board's September 28, 2020 Decision—which maintained in the '1714 proceeding rejections from the '1697 proceeding that must be terminated—should be reconsidered and modified. That decision is flatly contrary to the unambiguous command of Section 317(b), as explained by the Federal Circuit in its binding *Fairchild* and *VirnetX* decisions. Because the Board has "convincing[ly] err[ed]" and/or "abuse[d] discretion" in maintaining the estopped rejections, the Board should grant this request to review and reconsider its earlier decision. MPEP § 2681 (IV).
>
> In relevant part, pre-AIA Section 317(b) unambiguously mandates:
>
> Once a final decision has been entered against a party in a civil action arising in whole or in part under section 1338 of title 28, that the party has not sustained its burden of proving the invalidity of any patent claim in suit . . . , then neither that party nor its privies may thereafter request an *inter partes* reexamination of any such patent claim on the basis of issues which that party or its privies raised or could have raised in such civil action or *inter partes* reexamination

> proceeding, and an *inter partes* reexamination requested by that party or its privies on the basis of such issues may not thereafter be maintained by the Office, notwithstanding any other provision of this chapter.

Id., at 8.

Patent Owner also asserts that:

> the Board's insistence that the October 16, 2019 Petition Decision "only ordered a new RAN be mailed in the '1697 reexamination proceeding, [and] not in the '1714 reexamination proceeding," (September 28, 2020 Decision at 11), is beside the point. In the October 16, 2019 Petition Decision, the Office correctly recognized that Section 317(b) prohibited maintaining reexamination of claims 1-6 and 13-16 of the '151 patent based on the Apple-proposed rejections. The Board's September 28, 2020 decision is rooted in the faulty premise that the merged reexamination proceedings were properly maintained until they were finally severed in October 2019, and that the Board, in an appeal from the Examiner's decision in the '1714 proceeding, can continue to consider the estopped rejections proposed in the '1697 proceeding. But if the Office had properly granted VirnetX's original petition to sever the two proceedings and to terminate Apple's proceeding with respect to the claims subject to Section 317(b)'s estoppel, the estopped rejections would have never been part of the Examiner's RAN, nor remained before the Board.

Id., at 10-11.

Patent Owner asserts in Footnote 2:

> For that reason, the Board's reliance on *In re Affinity Labs of Texas, LLC*, 856 F.3d 883 (Fed. Cir. 2017), (see September 28, 2020 Decision at 9-10), is entirely misplaced. *Affinity Labs* held that Section 317(b)'s estoppel applies only to the "requester that was a party to the civil action or its privies"—but not to other, unrelated parties. 856 F.3d at 893. Here, *Affinity Labs* would have required termination of the reexamination requested by Apple with respect to the upheld claims as of September 2014—the date on which the

> Federal Circuit affirmed the district court's validity ruling. Nothing in *Affinity Labs* permits the Office to continue to consider those rejections simply because the estopped reexamination proceeding is merged with another proceeding where a different requester proposed different rejections.

Id., at 11.

Patent Owner contends that:

> Finally, the September 28, 2020 Decision invokes the Examiner's authority to consider '"the best references in his or her command,"' including "'patents and printed publications ... cited by another reexamination requester.'" (September 28, 2020 Decision at 11-12 (quoting 37 C.F.R. § 1.937, MPEP § 2656(A)).) As an initial matter, these regulatory provisions cannot override the plain statutory command of Section 317(b). Moreover, these provisions simply explain that the Examiner is not limited to the prior art asserted by the requester, but may consider the "patents and printed publications" that may have been cited elsewhere. They do not authorize the Examiner to rely on rejections, arguments, and evidence from the proceeding that is terminated by operation of Section 317(b). That is particularly so in this proceeding, where Apple introduced extensive briefing and evidence in the merged reexamination proceedings well after VirnetX's petition to sever and terminate. (See VirnetX's June 29, 2020 Petition To Vacate June 23, 2020 Board Decision at 7 & n.4.) None of Apple's arguments or evidence should have been part of those proceedings because VirnetX's petition to sever/terminate should have been granted at the outset. To treat these provisions as somehow permitting the Examiner to continue considering those rejections, arguments, and evidence in the Cisco-initiated '1714 proceeding is to render Section 317(b) a nullity.
>
> Furthermore, under pre-AIA Sections 312 and 313 applicable to this reexamination proceeding, the Board's consideration is limited to the resolution of the "substantial new question of patentability" identified by the Director when instituting review. 35 U.S.C. § 312(a) (2006); *see also id.* § 313 (2006).1 Thus, pre-AIA Section 312 provides that, in deciding whether to institute reexamination, "the Director shall determine whether a substantial new question of patentability affecting any claim of the patent concerned is raised

> by the request." 35 U.S.C. § 312(a) (2006) (emphasis added). And pre-AIA Section 313 provides, in turn, that "[i]f, in a determination made under section 312(a), the Director finds that a substantial new question of patentability affecting a claim of a patent is raised, the determination shall include an order for inter partes reexamination of the patent for resolution of the question." 35 U.S.C. § 313 (2006) (emphasis added). Here, the Director's order granting Cisco's request in the '1714 reexamination proceeding did not refer to Apple's proposed rejections or the prior art proposed by Apple; the order identified "a substantial new question of patentability" based solely on the prior art and rejections advanced by Cisco.

Id., at 11–13.

Patent Owner filed a renewed petition on June 2, 2017, that "renews its request for severance of *inter partes* reexamination control nos. 95/001,714 ("the '1,714 proceeding") and 95/001,697 ("the '1,697 proceeding) and termination as to claims 1-6 and 13-16 of U.S. Patent No. 7,490,151 ("the '151 patent") in the '1,697 proceeding. This relief is appropriate under 35 U.S.C. §317(b) (2006) and the recent decision by the United States Court of Appeals for the Federal Circuit in *Fairchild (Taiwan) Corp. v. Power Integrations, Inc.*, 854 F.3d 1364 (Fed. Cir. 2017)." A petition decision was mailed on October 16, 2019, which granted Patent Owner's renewed petition and the specific relief requested, i.e., severance of *inter partes* reexamination control nos. 95/001,714 and 95/001,697, and termination as to claims 1–6 and 13–16 of U.S. Patent No. 7,490,151 ("the '151 patent") in the '1,697 proceeding.

It is undisputed that Pre-AIA section 317(b) imposes estoppel limitations on a requester that was a party to the civil action or its privies. The Federal Circuit's construction of that statutory provision in *Fairchild*, 854 F.3d 1364, and *VirnetX Inc.*, 931 F.3d 1363, clearly support termination of the '1697 proceeding requested by Apple due to estoppel. However, in the related litigation, the Final Judgment of February 28, 2013, states:

> "On the eve of trial, only two parties remained: Apple and Cisco Systems, Inc. By agreement, the Court granted Defendants' Motion for Separate Trials, setting only Apple for trial starting October 31, 2012 (Docket No. 542). Since all issues, between VirnetX and Apple, except future ongoing royalties, if any, have been finally resolved either by the jury or the Court's Memorandum Opinion and Order (Docket No. 732), there is no reason to delay entering judgment as to Apple.
>
> Therefore, pursuant to Rule 54(b) of the Federal Rules of Civil Procedure, consistent with the Court's Memorandum Opinion and Order, and the Court having expressly determined that there is no just cause for delay, the Court ORDERS AND ENTERS FINAL JUDGMENT AS TO APPLE . ." *Virnetx Inc., v. Cisco Systems Inc.*, Case No. 6:10-CV-417 (E.D. TEX.)

Clearly, Cisco was not a party to this Final Judgment of February 28, 2013. However, to the extent that Patent Owner is suggesting again that the estoppel not only applies to Apple in the '1697 proceeding, but also extends to Cisco and the issues in the '1714 proceeding, contrary to Patent Owner's assertions, the *Fairchild* and *VirnetX* decisions do not support this notion. Also MPEP § 2686.04, Part IV states in pertinent part that "[t]he issuance of a final court decision **upholding validity** during an *inter partes* reexamination, where the person who filed the request **was not a party to the litigation**, will have no binding effect on the examination of the reexamination. This is because the court stated in *Ethicon v. Quigg,* 849 F.2d 1422, 1428, 7 USPQ2d 1152, 1157 (Fed. Cir. 1988) that the Office is not bound by a court's decision upholding validity and should continue the reexamination." Further, the plain language of pre-AIA section 317(b) does not permit extending the reach of estoppel as far as Patent Owner suggests.

As pointed out previously, the Federal Circuit rejected this notion in *In re: Affinity Labs* 856 F.3d 883, 893 (Fed. Cir. 2017) (emphasis added) (internal citations omitted):

> We also find no basis in the statute for Affinity's argument that the final decision in the Volkswagen litigation should have preclusive effect on the reexaminations requested by King or Apple. By its plain and unambiguous terms, pre-AIA section 317(b) . . . **imposes the aforementioned limitations only on a requester that was a party to the civil action or its privies. Because Apple was neither a party to the Volkswagen litigation nor was there any evidence Apple was Volkswagen's privy, we also find no error in the PTO's decision not to terminate the *inter partes* reexamination requested by Apple.**
>
> Affinity's invocation of the general policy goal of preventing abusive reexamination practices cannot override the statute. . . . The plain language of the statute does not permit extending the reach of estoppel as far as Affinity suggests. Moreover, we have reviewed the legislative history and find no such extraordinary showing sufficient to overcome the statute's plain language.

The petition decision of October 16, 2019, which granted severance of the merged reexaminations, sets forth the procedure to be followed subsequent to dissolution of the merger, and states in pertinent part:

> Patent owner's June 2, 2017 renewed petition under 37 CFR 1.182 to sever the merger of 95/001,697 and 95/001,714 (the merged proceeding), **and terminate reexamination of claims 1-6 and 13-16** of U.S. Patent No. 7,490,151 (the '151 patent) **in *inter partes* reexamination proceeding control number 95/001,697** (the '1697 proceeding) is granted.
>
> The merged proceeding of *inter partes* reexamination control numbers 95/001,697 and 95/001,714 is hereby severed.
>
> Prosecution of claims 1-16 of the '151 patent in *inter partes* reexamination control number 95/001,714 will continue.
>
> **The estoppel provisions of pre-AIA 35 U.S.C. 317(b) apply to any rejection of claims 1-6 and 13-16 of the '151 patent in the**

> **’1697 proceeding.** Any rejection which is presently applied against claims 1-6 and 13-16 of the ’151 patent, i.e., the patent under reexamination, in the ’1697 reexamination proceeding will not be further maintained by the Office. **No further rejection of claims 1-6 and 13-16 of the ’151 patent will be made in the ’1697 reexamination proceeding.**
>
> **The determination in this decision not to maintain, in the ’1697 reexamination proceeding, any rejection of claims 1-6 and 13-16 of the ’151 patent is made pursuant to the estoppel provisions of pre-AIA 35 U.S.C. 317(b),** and is not a “non-adoption of” or a “determination not to make” such a rejection within the meaning of 37 CFR 41.61. **For this reason, any notice of appeal or cross-appeal of the present determination in the ’1697 reexamination proceeding not to make or maintain a rejection of claims 1-6 and 13-16 of the ’151 patent will be held to be defective.**
>
> Prosecution in the ’1697 reexamination proceeding with respect to claims 7-12 of the ’151 patent will continue. The ’1697 reexamination proceeding will be forwarded to the Central Reexamination Unit (CRU) for processing and issuance of a new Right of Appeal Notice (RAN) consistent with this decision.

Renewed Pet. Dec. 2 (emphasis added) (emphasis omitted).

With regard to merger of reexamination proceedings, MPEP 2686.04 states in pertinent part:

> A decision to merge the reexamination proceedings will require that responses/comments by the patent owner and the third party requester(s) must consist of a single response/comment paper, addressed to both files, filed in duplicate each bearing a signature, for entry in both files. The same applies to any other paper filed in the merged proceeding. **The merger decision also will point out that both files will be maintained as separate complete files**. (emphasis added)

The Office followed its procedure set forth in the petition decision of October 16, 2019, which dissolved the merger. Although the ’1697 proceeding was severed from the instant appeal subsequent to the RAN, the

Petition Decision ordered that "[p]rosecution of claims 1–16 of the '151 patent in *inter partes* reexamination control number 95/001,714 will continue." Renewed Pet. Dec. 2, 14. The Petition Decision only ordered that a new RAN be mailed in the '1697 reexamination proceeding, not in the '1714 reexamination proceeding. *Id.* Therefore, Patent Owner's assertion that the Board exceeded its statutory authority because the Board's Decision in the '1714 reexamination decided the merits of rejections, which were not permitted to be applied against claims 1–6 and 13–16 of the '151 patent in the '1697 reexamination proceeding, is not consistent with the Petition Decision. The Petition Decision only explains how the estoppel provision will be applied to rejections in the issuance of a new RAN in the '1697 reexamination proceeding requested by Apple. Patent Owner's assertion is also not consistent with MPEP § 2686.04, Part IV, for the reasons discussed *supra*.

In a merged reexamination proceeding, both reexamination files are maintained as separate, complete files per Office procedure. If severance of a merged reexamination proceeding occurs, the Office includes specific procedural instructions in the decision to sever. The Office followed its own procedure by continuing prosecution of claims 1–16 in the '1714 proceeding, as directed in the decision to sever. Renewed Pet. Dec. 2, 14, 15.

The appeal of claims 1–16 in the '1714 proceeding is based on the rejections of claims 1–16 (and non-adoption of proposed rejections), set forth by the Examiner in the RAN of February 24, 2016. Once the Office adopts proposed rejections, the rejections are the Examiner's rejections, i.e., the position of the Office. Indeed, in rejecting claims in *inter partes* reexamination, the Examiner is not limited only to the prior art or rationales proposed by a requesting party, in this case Cisco. *See* 37 C.F.R. §§ 1.937 ("The *inter partes* reexamination proceeding will be conducted in accordance with §§ 1.104 through 1.116. . . ."), 1.104 ("In rejecting claims for want of novelty or for obviousness, the examiner must cite the best references at his or her command."); MPEP § 2656 (While the request for *inter partes* reexamination will be "the primary source of prior art[,] . . . the examiner must also consider patents and printed publications: (A) cited by another reexamination requester under 37 C.F.R. 1.510 or 37 C.F.R. 1.915; (B) cited by the patent owner under a duty of disclosure (37 CFR 1.933) in compliance with 37 CFR 1.98; (C) discovered by the examiner in searching; (D) of record in the patent file from earlier examination; . . .").

Nor is it inconsistent with 35 U.S.C. § 317(b) for the Examiner to maintain rejections over the best references at his or her command. 35 U.S.C. § 317(b) only requires actions consistent with the petition decision severing the merged reexaminations, and does not require withdrawal of any specific rejections maintained by the Examiner in the '1714 reexamination filed by Cisco. Patent Owner's speculation as to what may have occurred during prosecution had its petition to sever been granted earlier is hypothetical, and of no moment. It is noted that with regard to general procedure in an *inter partes* reexamination proceeding, if an Examiner withdraws a grounds of rejection, then with its next opportunity to file comments, the Requester may propose such grounds of rejection as a new grounds of rejection proposed by the Requester. *See* MPEP 2671.01 Part IV.

Also, the Requester may appeal any of its proposed rejections which are non-adopted rejections or withdrawn rejections, in accordance with 37 CFR 41.67(c)(1)(vi) and MPEP 2671.02 Part III.

A review of the Board's Decision for procedural matters, only to the extent of determining whether it involves a convincing showing of error, an abuse of discretion, or a policy issue reveals no error. The petition requesting reconsideration of the Board's decision of September 28, 2020, is **granted** to the extent that the petition decision has been reviewed, but the petition is **denied** because there is no convincing showing of error.

DECISION

In view of the foregoing, Patent Owner's petition is DENIED.

Patent Owner's request to reopen prosecution under 37 C.F.R. § 41.77(b)(1) of July 23, 2020, and Requester's Comments under 37 C.F.R. § 41.77(c) of August 21, 2020, are before the Board and will be considered in due course.

/s/ Scott C. Weidenfeller
Scott C. Weidenfeller
Vice Chief Administrative Patent Judge

PATENT OWNER:

PAUL HASTINGS LLP
2050 M STREET, NW
WASHINGTON, DC 20036

THIRD-PARTY REQUESTER:

SIDLEY AUSTIN, LLP
2021 MCKINNEY AVENUE, SUITE 2000
DALLAS, TX 75201

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/001,714	08/16/2011	7,490,151 B2	43614.99	3428

7590 03/24/2022

PAUL HASTINGS LLP
2050 M STREET, NW
WASHINGTON, DC 20036

EXAMINER	
RALIS, STEPHEN J	
ART UNIT	PAPER NUMBER
3992	
MAIL DATE	DELIVERY MODE
03/24/2022	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.
Requester and Cross-Appellant

v.

Patent of VIRNETX INC.
Patent Owner and Appellant

Appeal 2022-001473
Reexamination Control 95/001,714
Patent 7,490,151 B2
Technology Center 3900

Before JEFFREY B. ROBERTSON, DENISE M. POTHIER, and JEREMY J. CURCURI, *Administrative Patent Judges*.

ROBERTSON, *Administrative Patent Judge*.

DECISION UNDER 37 C.F.R. § 41.77(f)

This is a decision under 37 C.F.R. § 41.77(f). In a Decision on Appeal entered June 23, 2020 (Appeal No. 2020-000639, hereinafter "Decision"), the Board entered a new ground of rejection under 37 C.F.R. § 41.77(b) for claims 6 and 12. Decision 58–59, 62.[1] Specifically, the Decision reversed the Examiner's decision not to adopt the rejection of claims 6 and 12 as obvious over Aziz[2] and Edwards.[3] *Id.*

In response to the Decision, Patent Owner VirnetX, Inc. ("Patent Owner") requested reopening of prosecution under 37 C.F.R. § 41.77(b)(1). REQUEST TO REOPEN PROSECUTION filed July 23, 2020 (hereinafter "Request to Reopen" or "Req. to Reopen"). The Request to Reopen presented a Declaration of Dr. Fabian Monrose dated July 22, 2020 (hereinafter "2020 Monrose Declaration"), as evidence in response to the new grounds of rejection. Req. to Reopen 2.

Third Party Requester Cisco Systems, Inc. ("Cisco") filed THIRD PARTY COMMENTS TO PATENT OWNER'S REQUEST TO REOPEN PROSECUTION on August 21, 2020 (hereinafter "41.77(c) Comments" or "41.77(c) Comm.") in response to the Patent Owner's Request to Reopen.

[1] Patent Owner filed a petition to vacate the Decision on June 29, 2020, which was dismissed in a Decision on Petition entered on August 17, 2020. Patent Owner also filed a request for rehearing before the Precedential Opinion Panel or review by the Chief Judge of the August 17, 2020 Petition Decision on October 13, 2020, which was also denied in a Decision on Petition entered April 27, 2021.

[2] Ashar Aziz et al., U.S. Patent No. 6,119,234, issued September 12, 2000 ("Aziz").

[3] Nigel Edwards & Owen Rees, *High security Web servers and gateways*, 29 Computer Networks and ISDN Systems 927–938 (Sept. 1997) ("Edwards").

In an order remanding the case to the Examiner, Patent Owner's Request to Reopen was entered for consideration by the Examiner. ORDER REMANDING INTER PARTES REEXAMINATION UNDER 37 C.F.R. § 41.77(d) to the EXAMINER entered June 22, 2021. In addition, Requesters' comments were also entered for consideration by the Examiner.

An Examiner's Determination under 37 C.F.R. § 41.77(d) was issued on August 17, 2021 (hereinafter, "Ex. Det."), in which the Examiner determined that Patent Owner's Response was insufficient to overcome the new rejection set forth in the Decision. Ex. Det. 5–6.

Patent Owner filed a response to the Examiner's Determination. PATENT OWNER'S COMMENTS IN RESPONSE TO EXAMINER'S DETERMINATION PURSUANT TO 37 C.F.R. § 41.77(e) filed September 17, 2021 (hereinafter, "PO 41.77(e) Resp."). Cisco filed a reply to Patent Owner's 37 C.F.R. § 41.77(e) Response. THIRD PARTY REQUESTER'S REPLY TO PATENT OWNER'S COMMENTS IN RESPONSE TO EXAMINER'S DETERMINATION PURSUANT TO 37 C.F.R. § 41.77(e) filed October 5, 2021, (hereinafter "Req. 41.77(e) Rep.").

Pursuant to 37 C.F.R. § 41.77(f) the proceeding has been returned to the Board in order to "reconsider the matter and issue a new decision." Also pursuant to 37 C.F.R. § 41.77(f), this decision is deemed to incorporate the earlier decision, except as for those portions specifically withdrawn.

Discussion

Claim 6 of United States Patent 7,490,151 B2 (hereinafter the "'151 Patent"), depends from claim 1 and further recites "wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client." Cisco's Appeal Brief filed June 3, 2016 (hereinafter "Cisco Appeal Br.") 18. Claim 12 depends from claim 7, and recites similar limitations. *Id.* at 19.

In the Decision, we determined that the combination of Edwards, which discloses a naming service interceptor that prevents services in the internal network from accidently subverting security, and Aziz, which discloses including automatically initiating the encrypted channel between the client and the secure server, renders obvious avoiding sending a true IP address of the secure server to the client as recited in claim 6. Decision 59, citing Edwards 932, 936; Aziz, col. 11, ll. 64–65; *see also* Decision 29 (discussing that Aziz discloses automatically initiating an encrypted channel between the client and the secure server).

Patent Owner contends that the Decision misunderstood the teachings of the references. Req. to Reopen 2. Patent Owner contends that Aziz and Edwards do not render obvious the full limitation "wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client" recited in claim 6. *Id.* at 3. Patent Owner argues Edwards discloses the web server, not the client, includes a CORBAweb plugin and that an interceptor receives requests from the CORBAweb plugin. *Id.* at 3–4, citing 2020 Monrose Declaration ¶¶ 18–21. Patent Owner argues that if Aziz's system were modified so that the client only received an address of a service interceptor, there is no evidence of how an encrypted channel would extend between the client and secure server instead of between the client and the service interceptor. *Id.* at 4, citing Monrose Declaration ¶ 22.

Cisco contends Edwards discloses that services are available to the client only through the name service interceptor and proxy such that the true identification is not provided to the client. 41.77(c) Comm. 2–3, citing Edwards 936. Cisco argues Patent Owner has not addressed the language of the claims by focusing on what is provided to the client in Edwards, rather than what is not provided to the client as recited in claim 6. *Id.* at 3. Cisco contends Aziz and Edwards are properly combinable, where Patent Owner only reiterates arguments that the Board has already rejected. *Id.* at 3–5. Cisco contends Patent Owner's arguments are against bodily incorporation of Aziz and Edwards, rather than the combined disclosures of the prior art. *Id.* at 5.

The Examiner determined that the 2020 Monrose Declaration failed to overcome the new ground of rejection in the Decision. Ex. Det. 5. The Examiner found Edwards discloses a proxy providing a reference to a service interceptor instead of an actual service, such that Edwards teaches or renders obvious avoiding sending a true IP address of the secure server to the client. *Id.* As to Patent Owner's position that it would not have been obvious to have modified Aziz with Edwards to arrive at the subject matter of claim 6, the Examiner determined that Patent Owner had unduly focused on Edwards rather than the combination of Aziz and Edwards. *Id.* at 5–6.

Because the 2020 Monrose Declaration constitutes the new evidence to be considered in response to the rejection of claims 6 and 12 under 37 C.F.R. 41.77(b)(1), for the purposes of this decision, we focus on our discussion on the 2020 Monrose Declaration. The 2020 Monrose Declaration states that "Edwards does not provide any disclosure regarding whether the client, or the web server, is provided with the address of the service." 2020 Monrose Declaration. ¶ 21. The 2020 Monrose Declaration states that "it is my opinion that a person of ordinary skill in the art would not have been motivated to combine the disclosures of Aziz and Edwards in a manner that would have met the limitations of claims 6 and 12" because "it would have been unclear to a person of ordinary skill in the art how, in the context of Aziz, an encrypted channel would extend 'between the client and the secure server' instead of between the client and service interceptor." *Id.* at ¶ 22. The 2020 Monrose Declaration opines that there is no guidance or description in Aziz and Edwards of how or why such a combination would have met the limitations of claims 6 and 12. *Id.* Thus, the 2020 Monrose Declaration expresses the opinion that a person of ordinary skill in the art would not have had a reason to implement the suggested combination or have an expectation that such a combination would have been successful. *Id.*

We are not persuaded by the 2020 Monrose Declaration. As discussed by Cisco and the Examiner, the 2020 Monrose Declaration discusses the individual teachings of each reference and bodily incorporation thereof. 41.77(c) Comm. 5; Ex. Det. 5–6. Indeed, the 2020 Monrose Declaration only discusses certain disclosures in Edwards to support the position that Edwards discloses the web server, and not the client, includes the CORBAweb plugin. 2020 Monrose Declaration ¶¶ 18–21, citing Edwards 930, 932, 936, Figs. 3, 4. In this regard, the 2020 Monrose

Declaration acknowledges Edwards discloses an interceptor for a name service that returns a reference to a service interceptor instead of the actual service. *Id.* However, the 2020 Monrose Declaration does not sufficiently explain why any lack of disclosure in Edwards as to whether the client or web server is provided with address of the service would mean that when the combination of Aziz and Edwards is considered as a whole, one of ordinary skill in the art would not have considered the recitation that establishing an encrypted channel between the client and secure server "avoids sending a true IP address of the secure server to the client" as recited in claims 6 and 12.

In this regard, Aziz specifically discloses a desire to hide network topology in conjunction with automatically establishing an encrypted channel between the client and secures server. Decision 28–29 (discussing that Aziz discloses an application 215 running on an authorized client 210 that makes a query in the form of a DNS request for a secure server received by resolver 225, which results in automatically initiating an encrypted channel between the client and secure server), 58–59; Aziz, col. 11, ll. 64–65. Edwards discloses the use of service interceptors to prevent the accidental subverting of security. Edwards 936. In addition, Edwards emphasizes the flexibility of CORBA for various processes. *Id.* at 931.

The Supreme Court in *KSR* has stated "[a]s our precedents make clear, however, the analysis [under 35 U.S.C. § 103] need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *KSR Int'l. Co. v. Teleflex Inc.*, 550

U.S. 398, 418 (2007). "A person of ordinary skill is also a person of ordinary creativity, not an automaton." *Id*. at 421.

In this case, the 2020 Monrose Declaration expresses opinions based upon the lack of express teachings and guidance in Aziz and Edwards without sufficiently addressing why the proposed combination would have been beyond the inferences and creative steps employed by one of ordinary skill in the art. That is, the 2020 Monrose Declaration indicates that the level of ordinary skill is high, including a master's degree in computer science or computer engineering with two years of experience in computer networking with some accompanying exposure to network security. 2020 Monrose Declaration ¶ 4. Accepting this description of the level of skill in the art, we are of the view that more explanation is necessary beyond any lack of express teachings in Aziz and Edwards as to the recitations in claims 6 and 12 in order to support the position that the combination of Aziz and Edwards as a whole fail to render claims 6 and 12 obvious. It is unclear from the 2020 Monrose Declaration why, when combining Aziz and Edwards, the result would necessarily be only an encrypted channel between the client and a service interceptor, and why one of ordinary skill in the art would have had an issue with establishing an encrypted channel between the client and the secure server that avoids sending the true IP address of the secure server to the client.

It is well settled that the Board is entitled to weigh declarations expressing opinions as to fact and conclude that the lack of factual corroboration warrants discounting the opinions expressed in the declarations. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1368 (Fed.

Cir. 2004); *cf. In re Thompson*, 545 F.2d 1290, 1295 (CCPA 1976) (affidavit fails if it recites conclusions without reciting the underlying facts to support those conclusions). In this case, although we assign weight to the opinions made in the 2020 Monrose Declaration, we find that there is insufficient explanatory support therein as to why a person of ordinary skill in the art would not have been capable of the modifications suggested by the disclosures of Aziz and Edwards to outweigh the evidence in favor of obviousness for the reasons expressed above and in the Decision.

Lastly, we do not subscribe to Patent Owner's view that the Examiner's Determination did not give due consideration to Patent Owner's arguments and evidence submitted in the Request to Reopen. PO 41.77(e) Resp. 3–5. Such procedural considerations are appropriately addressed by way of a timely filed petition. *See* 37 C.F.R. § 1.181. However, to the extent relevant, we agree with Cisco that the Examiner provided sufficient explanation in the Determination from which to ascertain that Patent Owner's evidence and arguments were considered. Req. 41.77(e) Resp. 2–3. The Examiner's Determination expressed the Examiner's view that the 2020 Monrose Declaration and Request to Reopen did not sufficiently address the combination of Aziz and Edwards as a whole, and as such, did not provide a sufficient basis to overcome the rejection of claims 6 and 12. Ex. Det. 5–6.

I. CONCLUSION

In view of the discussion above, we find Patent Owner's evidence and argument to be insufficient to overcome the new grounds of rejection under

37 C.F.R. § 41.77(b) of claims 6 and 12 under 35 U.S.C. § 103(a) as unpatentable over Aziz and Edwards as set forth in the Decision.

Thus, the status of the Adopted Rejections is as follows:

Claims Rejected	35 U.S.C. §	References/Basis	Affirmed	Reversed
1, 2, 5, 7, 8, 11, 13, 14	102(b)	Aventail Connect v3.01	13, 14	1, 2, 5, 7, 8, 11
1, 2, 5, 7, 8, 11, 13, 14	102(b)	AutoSOCKS	13, 14	1, 2, 5, 7, 8, 11
1, 2, 5, 7, 8, 11, 13, 14	103(a)	Beser, Kent	1, 2, 5, 7, 8, 11, 13, 14	
1–4, 7–10, 13–16	102(b)	Kiuchi	13, 14	1–4, 7–10, 15, 16
5, 11	103(a)	Kiuchi, Martin		5, 11
1, 7, 13	103(a)	Aziz, Edwards	1, 7, 13	
5, 11	103(a)	Aziz, Edwards, Martin	5, 11	
1–4, 6–10, 12–16	103(a)	Kiuchi, Edwards	13, 14	1–4, 6–10, 12, 15, 16
5, 11	103(a)	Kiuchi, Edwards, Martin		5, 11
Overall Outcome			1, 2, 5, 7, 8, 11, 13, 14	3, 4, 6, 9, 10, 12, 15, 16

The status of the Non-Adopted Rejections is as follows:

Claims	35 U.S.C. §	References/Basis	Affirmed	Reversed	New Ground
6, 12	102(b)	Kiuchi	6, 12		
1–4, 6–10, 12–16	102(e)	Wesinger	1–4, 6–10, 12–16		
5, 11	103(a)	Wesinger, Martin	5, 11		
1, 7, 13	102(e)	Blum	1, 7, 13		
2–4, 6, 8–10, 12, 14–16	103(a)	Aziz, Edwards	2–4, 8–10, 14–16	6, 12	6, 12
1–4, 6–10, 12–16	103(a)	Wesinger, Edwards	1–4, 6–10, 12–16		
5, 11	103(a)	Wesinger, Edwards, Martin	5, 11		
Overall Outcome			1–5, 7–11, 13–16	6, 12	6, 12

In accordance with 37 C.F.R. § 41.79(a)(4), the "[p]arties to the appeal may file a request for rehearing of the decision within one month of the date of: . . . [t]he new decision of the Board under § 41.77(f)." A request for rehearing must be in compliance with 37 C.F.R. § 41.79(b). Comments in opposition to the request and additional requests for rehearing must be in accordance with 37 C.F.R. § 41.79(c)–(d), respectively. Under 37 C.F.R. § 41.79(e), the times for requesting rehearing under paragraph (a) of this section, for requesting further rehearing under paragraph (c) of this section, and for submitting comments under paragraph (b) of this section may not be extended.

An appeal to the United States Court of Appeals for the Federal Circuit under 35 U.S.C. §§ 141–144 and 315 and 37 C.F.R. § 1.983 for an *inter partes* reexamination proceeding "commenced" on or after November

2, 2002 may not be taken "until all parties' rights to request rehearing have been exhausted, at which time the decision of the Board is final and appealable by any party to the appeal to the Board." 37 C.F.R. § 41.81. *See also* MPEP § 2682 (8th ed., Rev. 8, July 2010).

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a).

Requests for extensions of time in this *inter partes* reexamination proceeding are governed by 37 C.F.R. § 1.956. *See* 37 C.F.R. § 41.79.

In the event neither party files a request for rehearing within the time provided in 37 C.F.R. § 41.79, and this decision becomes final and appealable under 37 C.F.R. § 41.81, a party seeking judicial review must timely serve notice on the Director of the United States Patent and Trademark Office. *See* 37 C.F.R. §§ 90.1 and 1.983.

AFFIRMED; 37 C.F.R. § 41.77(f)

PATENT OWNER:

PAUL HASTINGS LLP
2050 M STREET, NW
Washington, DC 20036

THIRD PARTY REQUESTER:

HAYNES AND BOONE, LLP
IP SECTION
2323 VICTORY AVE., SUITE 700
DALLAS, TX 75219

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/001,714	08/16/2011	7,490,151 B2	43614.99	3428

7590 02/10/2023

PAUL HASTINGS LLP
2050 M STREET, NW
WASHINGTON, DC 20036

EXAMINER
RALIS, STEPHEN J

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
02/10/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.
Requester and Cross-Appellant

v.

Patent of VIRNETX INC.
Patent Owner and Appellant

Appeal 2022-001473
Reexamination Control 95/001,714
Patent 7,490,151 B2
Technology Center 3900

Before JEFFREY B. ROBERTSON, DENISE M. POTHIER, and JEREMY J. CURCURI, *Administrative Patent Judges*.

ROBERTSON, *Administrative Patent Judge*.

DECISION ON REHEARING UNDER 37 C.F.R. § 41.79

VirnetX Inc. ("Patent Owner"), the owner of the patent under reexamination (hereinafter "the '151 Patent"), requests rehearing of the Decision under 37 C.F.R. § 41.77(f) mailed March 24, 2022 (hereinafter "March 2022 Decision") as well as the Decision on Appeal mailed June 23, 2020 (hereinafter "2020 Decision"). Request for Rehearing filed April 25, 2022, hereinafter "PO Request" 1.

Patent Owner contends that the Board misapprehended or overlooked several arguments presented by Patent Owner. PO Request 1.

Substantively, Patent Owner identifies eight points misapprehended or overlooked by the Board as follows:

1. The Board deviated from the Examiner's position of mapping Kiuchi's client-side proxy to the claimed domain name server (DNS) proxy module, and the Board's mapping of Kiuchi's client-side proxy and C-HTTP name server to the DNS proxy module that performs steps (i) and (ii)[1] "violates the cardinal anticipation rule that the prior-reference must disclose 'all of the limitations arranged or combined in the same way as recited in the claim'"[2] because the client-side proxy and C-HTTP server are not part of the same module, and Kiuchi does not disclose forwarding the same DNS request it receives;
2. The Board's findings with respect to automatically initiating an encrypted channel in Kiuchi are inconsistent with the

[1] Steps (i) and (ii) are found, for example, in claim 13. Patent Owner Appeal Brief filed June 5, 2016, hereinafter "PO Appeal Br.," Claims App. iii.
[2] PO Request 3, (quoting *NetMoneyIN, Inc. v. Verisign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008) (emphasis omitted)).

Board's finding that the combination of the client-side proxy and C-HTTP name server disclosed in Kiuchi corresponds to the DNS proxy module that performs steps (i) and (ii);

3. In addressing the rejections based on Kiuchi and Edwards, the Board did not address any arguments directed to Edwards;
4. In addressing the rejections based on Beser and Kent, the Board overlooked Patent Owner's argument that the combination fails to disclose or suggest "a domain name server (DNS) proxy module that intercepts DNS requests sent by a client" and the Board failed to substantively address Patent Owner's arguments for other limitations;
5. The Board overlooked arguments as to the status of Aventail v3.01 and AutoSOCKS as prior art;
6. The Board failed to give proper consideration to Patent Owner's argument that "secure channel" should be construed as a "direct channel that is secure;"
7. The Board overlooked Patent Owner's argument that the Examiner had incorporated the proposed rejection in Cisco's request relying on outside name server (NS) 120 disclosed in Aziz in order to reject the claims; and
8. The Board did not make any fact finding that Aziz discloses receiving a request pertaining to a first entity at another entity.

PO Request 2–6.

Third Party Requester Cisco Systems, Inc. (hereinafter "Cisco") filed comments opposing Patent Owner's Request pursuant to 37 C.F.R.

§ 41.79(c). Third Party Requester's Comments In Response To Patent Owner's Request for Rehearing Pursuant to 37 C.F.R. § 41.79(c) filed on May 25, 2022, hereinafter "TPR Comm." Cisco contends that the Board did not misapprehend or overlook any aspects pertaining to the points raised by Patent Owner. TPR Comm. 2–10.

REHEARING STANDARD

We emphasize that according to 37 C.F.R. § 41.79(b)(1):

> The request for rehearing must state with particularity the points believed to have been misapprehended or overlooked in rendering the Board's opinion reflecting its decision. Arguments not raised in the briefs before the Board and evidence not previously relied upon in the briefs are not permitted in the request for rehearing except as permitted by paragraphs (b)(2) and (b)(3) of this section.

Section 41.79(b)(2) permits new argument based on a recent relevant decision of the Board or Federal Court. Section 41.79(b)(3) permits new argument responding to a new ground of rejection under 37 C.F.R. § 41.77(b).

Against this backdrop, we evaluate Patent Owner's request for rehearing.

Discussion

At the outset, we acknowledge that the '151 Patent appears to have expired.[3] Thus, the broadest reasonable claim interpretation standard no

[3] The '151 Patent claims to be a division of Application No. 09/504,783, filed February 15, 2000, where the term is indicated to be extended by 818

longer applies. *In re Rambus, Inc.*, 753 F.3d 1253, 1256 (Fed. Cir. 2014). The claims are therefore construed to have their ordinary and customary meaning, as would be understood by a person of ordinary skill in the art, at the time of the invention, in light of the language of the claims, the specification, and the prosecution history of record. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1313–17 (Fed. Cir. 2005) (en banc); *In re Rambus Inc.*, 694 F.3d 42, 46 (Fed. Cir. 2012) ("[T]he Board's review of the claims of an expired patent is similar to that of a district court's review."). We address this change in the claim interpretation to the extent necessary as discussed below.

We do not revisit Patent Owner's continued objections to the application of certain rejections originating from reexaminations originally requested by Apple, Inc. in response to Patent Owner's request for rehearing. PO Request 1–2, 2 n.1 (Patent Owner noting that it "reiterates" the argument "for preservation purposes in the event that this issue is deemed properly considered by the Board and not a petitionable issue"). This issue has been addressed multiple times by the Office,[4] and does not constitute a point misapprehended or overlooked in the 2020 Decision or the March 2022 Decision. *See* TPR Comm. 2–3.

days. *See* the '151 Patent 1 (*), (60). Thus, we understand the '151 Patent to have expired on May 14, 2022.

[4] *See* 2020 Decision 2 (FN1); DECISION ON PETITION mailed September 28, 2020; DECISION ON PETITION mailed April 27, 2021.

Point 1

We are not persuaded that the Board misapprehended or overlooked aspects identified by Patent Owner relating to the mapping of the DNS proxy module recited in claim 13 to Kiuchi.[5] In particular, we are not persuaded by Patent Owner's argument that the Board relied on a different position from the Examiner in finding that the combination of the client-side proxy and the C-HTTP name server corresponds to the recited DNS proxy module performing steps (i) and (ii) of claim 13. Patent Owner's position is without merit. As pointed out by Cisco and further discussed in the 2020 Decision, the Examiner's position throughout prosecution has been that the combination of Kiuchi's client-side proxy and C-HTTP name server perform the steps (i) and (ii) recited in claim 13. TPR Comm. 3–5; 2020 Decision 11 (citing Ans. 67–70, Chart E-1.1). Indeed, even the portions of Patent Owner's Appeal Brief cited in the Request for Rehearing acknowledge this position. PO Request 2–3 (citing PO Appeal Br. 35, 37–38 (discussing that the Request for Reexamination relied on the combination of Kiuchi's C-HTTP name server and client-side proxy to meet steps (i) and (ii) in claim 13)).

As to Patent Owner's argument that Kiuchi's client-side proxy and C-HTTP name server are not part of the "same module" and thus do not constitute a "DNS proxy module" (PO Request 3), we are not persuaded that the Board overlooked this argument. As previously pointed out by Cisco, Patent Owner has not provided any particular definition or claim

[5] The 2020 Decision reversed the rejection of independent claims 1 and 7 as anticipated by Kiuchi. *See* 2020 Decision 13.

interpretation that would distinguish the "module" recited in claim 13 from the arrangement in Kiuchi. *See* Cisco Respondent Brief filed August 18, 2016, 15.

We are unpersuaded that we misapprehended or overlooked Patent Owner's arguments alleging that Kiuchi generates a new DNS request for DNS lookup when a permission request to the C-HTTP name server fails. PO Request 3–4. To the extent Patent Owner's argument is that Kiuchi does not disclose DNS requests, we did not overlook this argument as discussed in the 2020 Decision. 2020 Decision 10. To the extent Patent Owner now argues that Kiuchi discloses DNS requests, but such requests are different than the original DNS request, we are of the view that such a new argument is untimely and not appropriate for a request for rehearing. *See* 37 C.F.R. § 41.79(b)(1). Substantively, Patent Owner's arguments are similar to its prior argument that Kiuchi's C-HTTP name server does not involve DNS. PO Request 3. In this regard, although Patent Owner cites to portions of Kiuchi (PO Request 3, (citing Kiuchi 64–65, 68)), Patent Owner does not sufficiently explain why these portions necessarily mean that new DNS requests are generated.

Point 2

Patent Owner contends the Board's finding that Kiuchi discloses "automatically initiating an encrypted channel between the client and the secure channel" as recited in claim 13 is inconsistent with the Board's finding that the combination of the client-side proxy and the C-HTTP name server correspond to the DNS proxy module, because the same DNS proxy module must perform all three claimed steps. PO Request 4. We agree with Cisco that Patent Owner has not identified any points misapprehended or overlooked, or for that matter, any inconsistency in the Board's findings. TPR Comm. 5–6. The 2020 Decision explains that the client-side proxy is part of the DNS proxy module and sends a request for initiating a secure channel between the client and secure server. *See* 2020 Decision 12–13. As such, it is unclear why Patent Owner believes there is an inconsistency in the Board's findings.

Point 3

As to Patent Owner's contention that for the rejections based on Kiuchi and Edwards, the Board did not address any of Patent Owner's arguments with respect to Edwards (PO Request 4), we are not persuaded that Patent Owner has identified any particular aspects of the rejection or argument that were misapprehended or overlooked. In the 2020 Decision, the Board observed that Patent Owner's arguments with respect to Edwards were based on the position that Edwards did not make up for deficiencies identified for Kiuchi. 2020 Decision 34 (citing PO Appeal Br. 61–62); 2020 Decision 31–32 (discussing Edwards's disclosure on interceptors); *see also* PO Appeal Br. 64 (acknowledging that Cisco and the Office only rely on Edwards for the position that it was well known for an intermediary module to intercept requests and arguing that Kiuchi fails to disclose automatically initiating an encrypted channel between the client and the secure server, where Edwards does not remedy the deficiencies of Kiuchi). Thus, the

Board considered Patent Owner's arguments regarding Edwards. *See* TPR Comm. 6–7.

Point 4

Patent Owner generally asserts that for the rejections involving Beser and Kent, the Board overlooked Patent Owner's argument that the combination thereof does not disclose a domain name server proxy module that intercepts DNS request sent by a client. PO Request 4. However, as Cisco points out, the Board addressed this argument in the 2020 Decision, and Patent Owner does not identify any particular points misapprehended or overlooked with respect to the 2020 Decision. TPR Comm. 7–8; 2020 Decision 42–43; PO Request 4.

As to Patent Owner's contention that the Board did not substantively engage with Patent Owner's arguments for other limitations, Patent Owner does not identify, with particularity, how the 2020 Decision misapprehended or overlooked Patent Owner's arguments by pointing out that Patent Owner had not addressed the Examiner's rationale. PO Request 4; 2020 Decision 43.

Point 5

Patent Owner contends that the Board overlooked arguments as to the prior art status of Aventail Connect v3.01 and AutoSOCKS, specifically, arguments that there were various issues with Mr. Chester's testimony and the Board improperly relied on testimony submitted by Apple. PO Request 5. We are not persuaded that the Board overlooked Patent Owner's arguments. The 2020 Decision expressly references the pages of Patent Owner's Appeal Brief identified by Patent Owner in the Request for Rehearing. *Compare* 2020 Decision 16 (citing PO Appeal Br. 4–7), *with* PO Request 5 (citing PO Appeal Br. 5–6). In the 2020 Decision, the Board explained with reference to the evidence of record why the evidence was sufficient to support the Examiner's position that Aventail Connect v3.01 and AutoSOCKS were publicly accessible and qualified as prior art. 2020 Decision 18–20, 23. Accordingly, the Board did not misapprehend or overlook Patent Owner's arguments.

Point 6

Regarding claims 13 and 14, Patent Owner contends that the Board did not consider Patent Owner's argument that "secure channel" should be interpreted as a "direct channel that is secure." PO Request 5 (citing 2020 Decision 21). Patent Owner argues the Board's position that Patent Owner did not sufficiently explain how a "direct" channel would distinguish Aventail Connect v3.01 and AutoSOCKS "defies the Federal Circuit's prior characterization of Aventail." PO Request 5. That is, Patent Owner contends the Federal Circuit found that Aventail Connect v3.01 discloses a system involving indirect communication between a client computer and an intermediate server, where the intermediate server relays data to a target computer. *Id.* (quoting *VirnetX Inc. v. Mangrove Partners Master Fund, Ltd.*, 778 F. App'x 897, 909 (Fed. Cir. 2019) ("*Mangrove Partners*")).

We are not persuaded that we misapprehended or overlooked the Federal Circuit's decision in *Mangrove Partners* in the 2020 Decision. In

Mangrove Partners, the Federal Circuit held with respect to U.S. Patent No. 6,502,135 ("the '135 Patent"), which shares a common specification with the '151 Patent, that under the broadest reasonable interpretation and based on prosecution history disclaimer, the claim phrase "VPN between the client computer and the target computer" requires direct communication between the client computer and target computer. *Mangrove Partners,* 778 F. App'x at 910; *see also* 2020 Decision 21. The 2020 Decision addresses the Federal Circuit's decision in detail, expressly pointing out that the Federal Circuit's discussion of Aventail Connect v.3.01 involved a "VPN between the client computer and the target computer" recited in claim 1 of the '135 Patent, whereas claim 13 of the '151 Patent more broadly recites a "secure channel between the client and the secure server." 2020 Decision 20–21. The Board explained that in view of the presence of intermediate devices including Internet Service Providers, firewalls, and routers, which according to Patent Owner still fall within the bounds of "direct" communication, there was insufficient basis to distinguish claim 13 from Aventail Connect v3.01 when claim 13's scope is determined in light of the Specification. *Id.* at 21–22; *see* PO Appeal Br. 16–17.

We are of the view that the expiration of the '151 Patent does not change the Board's analysis in the 2020 Decision. In the 2020 Decision, we construed "secure channel between the client and the secure server," under the broadest reasonable interpretation, as a secure connection that does not require encryption and which may include intervening structures. 2020 Dec. 13, 20–21. Under the *Phillips* standard noted above, our construction and analysis does not change.[6] The '151 Patent discloses "[i]t is desired for the communications to be secure, that is, immune to eavesdropping." '151 Patent, col. 1, ll. 34–35. Thus, construing a "secure channel between the client and the secure server" under its ordinary meaning in light the Specification as *Phillips* prescribes, "secure" does not require a VPN, and as such does not implicate the particular aspects of "direct" communication implied by the presence of a VPN determined by the Federal Circuit in *Mangrove Partners* as discussed above. 778 F. App'x at 909. Thus, the change in claim construction standard does not affect the Board's decision.

Point 7

Patent Owner contends the Board overlooked a portion of Patent Owner's argument that with respect to the rejections over Aziz and Edwards, the Examiner relied on a combination of resolver 225 and outside NS 120 as the DNS proxy module. PO Request 5–6.

In the 2020 Decision, we expressly discussed the Examiner's position that resolver 225 disclosed in Aziz corresponded to the recited DNS proxy module. 2020 Decision 24. With respect to outside NS 120, the Board explained that the Examiner did not rely on outside NS 120 itself for further limitations recited in claim 1; rather, the Examiner relied on the parallel

[6] We observe that in IPR2015-01047 (the IPR on the '151 Patent at issue in *Mangrove Partners*), under the broadest reasonable interpretation, the Board interpreted "'secure' . . . consistent[] with its plain and ordinary meaning" as "immune to eavesdropping," as opposed to "some particularized meaning." *Mangrove Partners Master Fund, Ltd. v. VirnetX Inc.*, IPR2015-01047, Paper No. 122 at 18 (PTAB July 14, 2020) (Final Written Decision) (citing the '151 Patent col. 1, ll. 34–35).

operation of resolver 225 with respect to NS 120 in discussing how resolver 225 corresponds to the DNS proxy module recited in claim 1. 2020 Decision 24–25 (citing Action Closing Prosecution (ACP) 68); Aziz, col. 9, ll. 54–56, col. 10, ll. 4–5, 36–39, 42–48, Figs. 3, 4A; *see also* ACP 65 (expressly discussing Patent Owner's argument also raised in the Request for Rehearing, which the Examiner addresses on ACP 68).

Thus, the Board did not misapprehend or overlook this portion of Patent Owner's argument.

Point 8

Patent Owner contends that although the Board determined that the broadest reasonable interpretation of "intercepting" includes receiving requests as disclosed in Aziz, the Board made no factual findings that Aziz discloses receiving a request pertaining to a first entity at another entity. Request 6.

We are of the view that the expiration of the '151 Patent does not substantially change the interpretation in the 2020 Decision of "intercepting a request" as "receiving a request pertaining to a first entity at another entity." 2020 Decision 29–30. In this regard, we observe that in arguing this limitation, Patent Owner implies that there is a difference between "receiving" and "intercepting" a request. PO Appeal Br. 52–53 (citing the declaration of Angelos D. Keromytis dated July 20, 2012, hereinafter "the Keromytis Decl.," ¶ 162). Yet, the Keromytis Declaration upon which Appellant relies, does not set forth an interpretation for the term "intercepting" under either standard. Keromytis Decl. ¶¶ 161, 162.

Independent claims 1, 7, and 13 of the '151 Patent recite "intercepting" a DNS request and an additional "determining" step. PO Appeal Br. Claims App. i–iv. *See Phillips*, 415 F.3d at 1314 ("the claims themselves provide substantial guidance as to the meaning of particular claim terms") (citing *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *ACTV, Inc. v. Walt Disney Co.*, 346 F.3d 1082, 1088 (Fed. Cir. 2003) ("the context of the surrounding words of the claim also must be considered in determining the ordinary and customary meaning of those terms."). In addition, the '151 Patent discloses "DNS proxy 2610 intercepts all DNS lookup functions from client 2605 and determines whether access to a secure site has been requested." *See* the '151 Patent, col. 37, ll. 60–62, Fig. 26.

Thus, applying *Phillips* does not change the claim construction of "intercepting" and as such, we construe "intercepting" to mean "receiving a request pertaining to a first entity at another entity." Accordingly, we are of the view that the '151 Patent having expired does not affect the interpretation of the claims or the application of art thereto.

We also discern no points misapprehended or overlooked by Patent Owner's arguments, as the 2020 Decision makes the express findings Patent Owner alleges are missing. 2020 Decision 28 ("Aziz discloses an application 215 running on authorized client 210 (client) makes a query (DNS request) for the address of an inside host 140, which is received by a resolver 225 (DNS proxy module) . . . . Aziz, col. 10, ll. 35–41, col. 8, ll. 19–25; Figs. 1, 2A.").

Thus, we have considered Patent Owner's request, but we decline to modify the Decision.

CONCLUSION

Outcome of Decision on Rehearing:

Claim(s) Rejected	35 U.S.C. §	Reference(s)/Basis	Denied	Granted
13, 14	102(b)	Aventail Connect v3.01	13, 14	
13, 14	102(b)	AutoSOCKS	13, 14	
1, 2, 5, 7, 8, 11, 13, 14	103(a)	Beser, Kent	1, 2, 5, 7, 8, 11, 13, 14	
13, 14	102(b)	Kiuchi	13, 14	
1, 7, 13	103(a)	Aziz, Edwards	1, 7, 13	
5, 11	103(a)	Aziz, Edwards, Martin	5, 11	
13, 14	103(a)	Kiuchi, Edwards	13, 14	
6, 12	103(a)	Aziz, Edwards	6, 12	
Overall Outcome			1, 2, 5–8, 11–14	

Final Outcome of Appeal after Rehearing:

Adopted Rejections:

Claims Rejected	35 U.S.C. §	References/Basis	Affirmed	Reversed
1, 2, 5, 7, 8, 11, 13, 14	102(b)	Aventail Connect v3.01	13, 14	1, 2, 5, 7, 8, 11
1, 2, 5, 7, 8, 11, 13, 14	102(b)	AutoSOCKS	13, 14	1, 2, 5, 7, 8, 11
1, 2, 5, 7, 8, 11, 13, 14	103(a)	Beser, Kent	1, 2, 5, 7, 8, 11, 13, 14	

1–4, 7–10, 13–16	102(b)	Kiuchi	13, 14	1–4, 7–10, 15, 16
5, 11	103(a)	Kiuchi, Martin		5, 11
1, 7, 13	103(a)	Aziz, Edwards	1, 7, 13	
5, 11	103(a)	Aziz, Edwards, Martin	5, 11	
1–4, 6–10, 12–16	103(a)	Kiuchi, Edwards	13, 14	1–4, 6–10, 12, 15, 16
5, 11	103(a)	Kiuchi, Edwards, Martin		5, 11
Overall Outcome			1, 2, 5, 7, 8, 11, 13, 14	3, 4, 6, 9, 10, 12, 15, 16

Non-Adopted Rejections:

Claims	35 U.S.C. §	References/Basis	Affirmed	Reversed	New Ground
6, 12	102(b)	Kiuchi	6, 12		
1–4, 6–10, 12–16	102(e)	Wesinger	1–4, 6–10, 12–16		
5, 11	103(a)	Wesinger, Martin	5, 11		
1, 7, 13	102(e)	Blum	1, 7, 13		
2–4, 6, 8–10, 12, 14–16	103(a)	Aziz, Edwards	2–4, 8–10, 14–16	6, 12	6, 12
1–4, 6–10, 12–16	103(a)	Wesinger, Edwards	1–4, 6–10, 12–16		
5, 11	103(a)	Wesinger, Edwards, Martin	5, 11		
Overall Outcome			1–5, 7–11, 13–16	6, 12	6, 12

DENIED

PATENT OWNER:

PAUL HASTINGS LLP
875 15th Street, NW
Washington, DC 20005

THIRD PARTY REQUESTER:

DAVID L. MCCOMBS
HAYNES AND BOONE, LLP
2323 Victory Avenue, Ste. 700
Dallas, Texas 75219

US007490151B2

(12) **United States Patent**
Munger et al.

(10) **Patent No.:** **US 7,490,151 B2**
(45) **Date of Patent:** **Feb. 10, 2009**

(54) **ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST**

(75) Inventors: **Edward Colby Munger**, Crownsville, MD (US); **Robert Dunham Short, III**, Leesburg, VA (US); **Victor Larson**, Fairfax, VA (US); **Michael Williamson**, South Riding, VA (US)

(73) Assignee: **Virnetx Inc.**, Scotts Valley Drive, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 818 days.

(21) Appl. No.: **10/259,494**

(22) Filed: **Sep. 30, 2002**

(65) **Prior Publication Data**

US 2003/0037142 A1 Feb. 20, 2003

Related U.S. Application Data

(60) Division of application No. 09/504,783, filed on Feb. 15, 2000, now Pat. No. 6,502,135, which is a continuation-in-part of application No. 09/429,643, filed on Oct. 29, 1999, now Pat. No. 7,010,604.

(60) Provisional application No. 60/137,704, filed on Jun. 7, 1999, provisional application No. 60/106,261, filed on Oct. 30, 1998.

(51) **Int. Cl.**
G06F 15/173 (2006.01)

(52) **U.S. Cl.** **709/225**; 709/229

(58) **Field of Classification Search** 709/217–225, 709/229; 713/201

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,933,846	A	6/1990	Humphrey et al.

(Continued)

FOREIGN PATENT DOCUMENTS

DE	199 24 575	12/1999

(Continued)

OTHER PUBLICATIONS

Search Report (dated Aug. 23, 2002), International Application No. PCT/US01/13260.

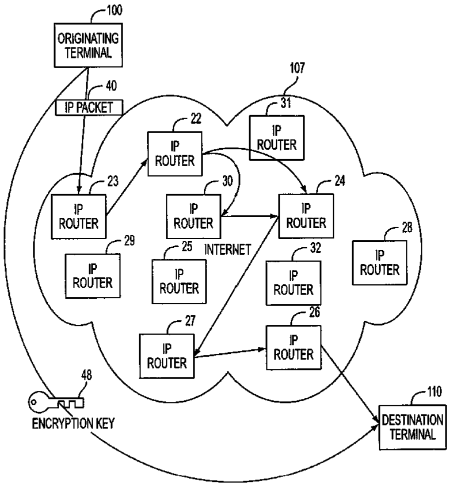
(Continued)

Primary Examiner—Krisna Lim
(74) *Attorney, Agent, or Firm*—McDermott Will & Emery

(57) **ABSTRACT**

A plurality of computer nodes communicate using seemingly random Internet Protocol source and destination addresses. Data packets matching criteria defined by a moving window of valid addresses are accepted for further processing, while those that do not meet the criteria are quickly rejected. Improvements to the basic design include (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities.

16 Claims, 35 Drawing Sheets



U.S. PATENT DOCUMENTS

4,988,990	A		1/1991	Warrior	
5,164,986	A	*	11/1992	Bright	380/273
5,276,735	A		1/1994	Boebert et al.	
5,311,593	A		5/1994	Carmi	
5,329,521	A		7/1994	Walsh et al.	
5,341,426	A		8/1994	Barney et al.	
5,367,643	A		11/1994	Chang et al.	
5,559,883	A		9/1996	Williams	
5,561,669	A		10/1996	Lenney et al.	
5,588,060	A		12/1996	Aziz	
5,625,626	A		4/1997	Umekita	
5,654,695	A		8/1997	Olnowich et al.	
5,682,480	A		10/1997	Nakagawa	
5,689,566	A		11/1997	Nguyen	
5,740,375	A		4/1998	Dunne et al.	
5,774,660	A		6/1998	Brendel et al.	
5,787,172	A		7/1998	Arnold	
5,790,548	A	*	8/1998	Sistanizadeh et al.	370/401
5,796,942	A		8/1998	Esbensen	
5,805,801	A		9/1998	Holloway et al.	
5,842,040	A		11/1998	Hughes et al.	
5,845,091	A		12/1998	Dunne et al.	
5,867,650	A		2/1999	Osterman	
5,870,610	A		2/1999	Beyda et al.	
5,878,231	A		3/1999	Baehr et al.	
5,892,903	A		4/1999	Klaus	
5,898,830	A	*	4/1999	Wesinger et al.	726/15
5,905,859	A		5/1999	Holloway et al.	
5,918,019	A		6/1999	Valencia	
5,996,016	A		11/1999	Thalheimer et al.	
6,006,259	A		12/1999	Adelman et al.	
6,006,272	A		12/1999	Aravamudan et al.	
6,016,318	A		1/2000	Tomoike	
6,016,512	A		1/2000	Huitema	
6,041,342	A		3/2000	Yamaguchi	
6,052,788	A		4/2000	Wesinger, Jr. et al.	
6,055,574	A		4/2000	Smorodinsky et al.	
6,061,736	A		5/2000	Rochberger et al.	
6,079,020	A	*	6/2000	Liu	713/201
6,092,200	A		7/2000	Muniyappa et al.	
6,101,182	A	*	8/2000	Sistanizadeh et al.	370/352
6,119,171	A		9/2000	Alkhatib	
6,119,234	A	*	9/2000	Aziz et al.	713/201
6,147,976	A		11/2000	Shand et al.	
6,157,957	A		12/2000	Berthaud	
6,158,011	A		12/2000	Chen et al.	
6,168,409	B1		1/2001	Fare	
6,175,867	B1		1/2001	Taghadoss	
6,178,409	B1		1/2001	Weber et al.	
6,178,505	B1		1/2001	Schneider et al.	
6,179,102	B1		1/2001	Weber et al.	
6,222,842	B1		4/2001	Sasyan et al.	
6,226,751	B1		5/2001	Arrow et al.	
6,233,618	B1		5/2001	Shannon	
6,243,360	B1		6/2001	Basilico	
6,243,749	B1		6/2001	Sitaraman et al.	
6,243,754	B1		6/2001	Guerin et al.	
6,256,671	B1	*	7/2001	Strentzsch et al.	709/227
6,263,445	B1		7/2001	Blumenau	
6,286,047	B1		9/2001	Ramanathan et al.	
6,301,223	B1		10/2001	Hrastar et al.	
6,308,274	B1		10/2001	Swift	
6,311,207	B1		10/2001	Mighdoll et al.	
6,324,161	B1		11/2001	Kirch	
6,330,562	B1		12/2001	Boden et al.	
6,332,158	B1	*	12/2001	Risley et al.	709/219
6,353,614	B1		3/2002	Borella et al.	
6,425,003	B1	*	7/2002	Herzog et al.	709/223
6,430,155	B1		8/2002	Davie et al.	
6,430,610	B1		8/2002	Carter	
6,487,598	B1		11/2002	Valencia	
6,502,135	B1	*	12/2002	Munger et al.	709/225
6,505,232	B1		1/2003	Mighdoll et al.	
6,510,154	B1		1/2003	Mayes et al.	
6,549,516	B1		4/2003	Albert et al.	
6,557,037	B1		4/2003	Provino	
6,571,296	B1		5/2003	Dillon	
6,571,338	B1		5/2003	Shaio et al.	
6,581,166	B1		6/2003	Hirst et al.	
6,606,708	B1	*	8/2003	Devine et al.	713/201
6,618,761	B2		9/2003	Munger et al.	
6,671,702	B2		12/2003	Kruglikov et al.	
6,687,551	B2		2/2004	Steindl	
6,714,970	B1		3/2004	Fiveash et al.	
6,717,949	B1		4/2004	Boden et al.	
6,751,738	B2	*	6/2004	Wesinger et al.	713/201
6,760,766	B1		7/2004	Sahlqvist	
6,826,616	B2		11/2004	Larson et al.	
6,839,759	B2		1/2005	Larson et al.	
7,010,604	B1		3/2006	Munger et al.	
7,133,930	B2		11/2006	Munger et al.	
7,188,180	B2		3/2007	Larson et al.	
7,197,563	B2		3/2007	Sheymov et al.	
2002/0004898	A1		1/2002	Droge	
2003/0196122	A1	*	10/2003	Wesinger et al.	713/201
2005/0055306	A1		3/2005	Miller et al.	
2006/0059337	A1	*	3/2006	Poyhonen et al.	713/165

FOREIGN PATENT DOCUMENTS

EP	0 814 589		12/1997
EP	0 814 589	A	12/1997
EP	0 838 930		4/1998
EP	0 838 930	A	4/1998
EP	836306	A1	4/1998
EP	0 858 189		8/1998
GB	2 317 792		4/1998
GB	2 317 792	A	4/1998
GB	2 334 181	A	8/1999
GB	2334181	A	8/1999
WO	9827783	A	6/1998
WO	WO 98/27783		6/1998
WO	WO 9827783	A	6/1998
WO	WO 98 55930		12/1998
WO	WO 98 59470		12/1998
WO	WO 99 38081		7/1999
WO	WO 99 48303		9/1999
WO	WO 00/17775		3/2000
WO	WO 00/70458		11/2000
WO	WO 01 50688		7/2001

OTHER PUBLICATIONS

Donald E. Eastlake, 3rd, "Domain Name System Security Extensions", Internet Draft, Apr. 1998, pp. 1-51.

D. B. Chapman et al., "Building Internet Firewalls", Nov. 1995, pp. 278-375.

P. Srisuresh et al., "DNA extensions to Network address Translators (DNS_ALG)", Internet Draft, Jul. 1998, pp. 1-27.

James E. Bellaire, "New Statement of Rules—Naming Internet Domains", Internet Newsgroup, Jul. 30, 1995, 1 page.

D. Clark, "US Calls for Private Domain-Name System", Computer Society, Aug. 1, 1998, pp. 22-25.

August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace", Computer & Security, vol. 17, No. 4, 1998, pp. 293-298.

Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, Jun. 21, 1997, 4 pages.

Search Report (dated Jun. 18, 2002), International Application No. PCT/US01/13260.

Search Report (dated Jun. 28, 2002), International Application No. PCT/US01/13261.

Donald E. Eastlake, "Domain Name System Security Extensions", DNS Security Working Group, Apr. 1998, 51 pages.

D. B. Chapman et al., “Building Internet Firewalls”, Nov. 1995, pp. 278-297 and pp. 351-375.

P. Srisuresh et al., “DNS extensions to Network Address Translators”, Jul. 1998, 27 pages.

Laurie Wells, “Security Icon”, Oct. 19, 1998, 1 page.

W. Stallings, “Cryptography And Network Security”, 2nd Edition, Chapter 13, IP Security, Jun. 8, 1998, pp. 399-440.

W. Stallings, “New Cryptography and Network Security Book”, Jun. 8, 1998, 3 pages.

Search Report (dated Aug. 20, 2002), International Application No. PCT/US01/04340.

Shree Murthy et al., “Congestion-Oriented Shortest Multipath Routing”, Proceedings of IEEE Infocom, 1996, pp. 1028-1036.

Jim Jones et al., “Distributed Denial of Service Attacks: Defenses”, Global Integrity Corporation, 2000, pp. 1-14.

Fasbender, Kesdogan, and Kubitz: “Variable and Scalable Security: Protection of Location Information in Mobile IP”, IEEE publication, 1996, pp. 963-967.

Laurie Wells (Lancasterbibelmail MSN COM); “Subject: Security Icon” Usenet Newsgroup, Oct. 19, 1998, XP002200606.

Davila J et al, “Implementation of Virtual Private Networks at the Transport Layer”, Information Security, Second International Workshop, ISW ’99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pp. 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www.springerlink.com/content/4uac0tb0heccma89/fulltext.pdf> (Abstract).

Alan 0. Frier et al., “The SSL Protocol Version 3.0”, Nov. 18, 1996, printed from http://www.netscape.com/eng/ssl13/ draft302.txt on Feb. 4, 2002, 56 pages.

Davila J et al, “Implementation of Virtual Private Networks at the Transport Layer”, Information Security, Second International Workshop, ISW’99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pp. 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www. springerlink.com/content/4uac0tb0hecoma89/fulltext.pdf>.

Dolev, Shlomi and Ostrovsky, Rafil, Efficient Anonymous Multicast and Reception (Extended Abstract), 16 pages.

F. Halsall, “Data Communications, Computer Networks and Open Systems”, Chapter 4, Protocol Basics, 1996, pp. 198-203.

Glossary for the Linux FreeS/WAN project, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/ doc/glossary.html on Feb. 21, 2002, 25 pages.

J. Gilmore, “Swan: Securing the Internet against Wiretapping”, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3.doc/rationale.html on Feb. 21, 2002, 4 pages.

Linux FreeS/WAN Index File, printed from http://liberty.freewan.org/freeswan trees/freeswan-1.3/doc/ on Feb. 21, 2002, 3 pages.

Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs—Research), Crowds: Anonymity for Web Transactions, pp. 1-23.

RFC 2401-Security Architecture for the Internet Protocol (RTP).

RFC 2543-SIP: Session Initiation Protocol (SIP or SIPS).

Rubin, Aviel D., Geer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), “Web Security Sourcebook”, pp. 82-94.

Search Report, IPER (dataed Nov. 13, 2002), International Application No. PCT/US01/04340.

Search Report, IPER (dated Feb. 6, 2002), International Application No. PCT/US01/13261.

Search Report, IPER (dated Jan. 14, 2003), International Application No. PCT/US01/13260.

Shankar, A.U. “A verified sliding window protocol with variable flow control”. Proceedings of ACM SIGCOMM conference on Communications architectures & protocols. pp. 84-91, ACM Press, NY,NY 1986.

* cited by examiner

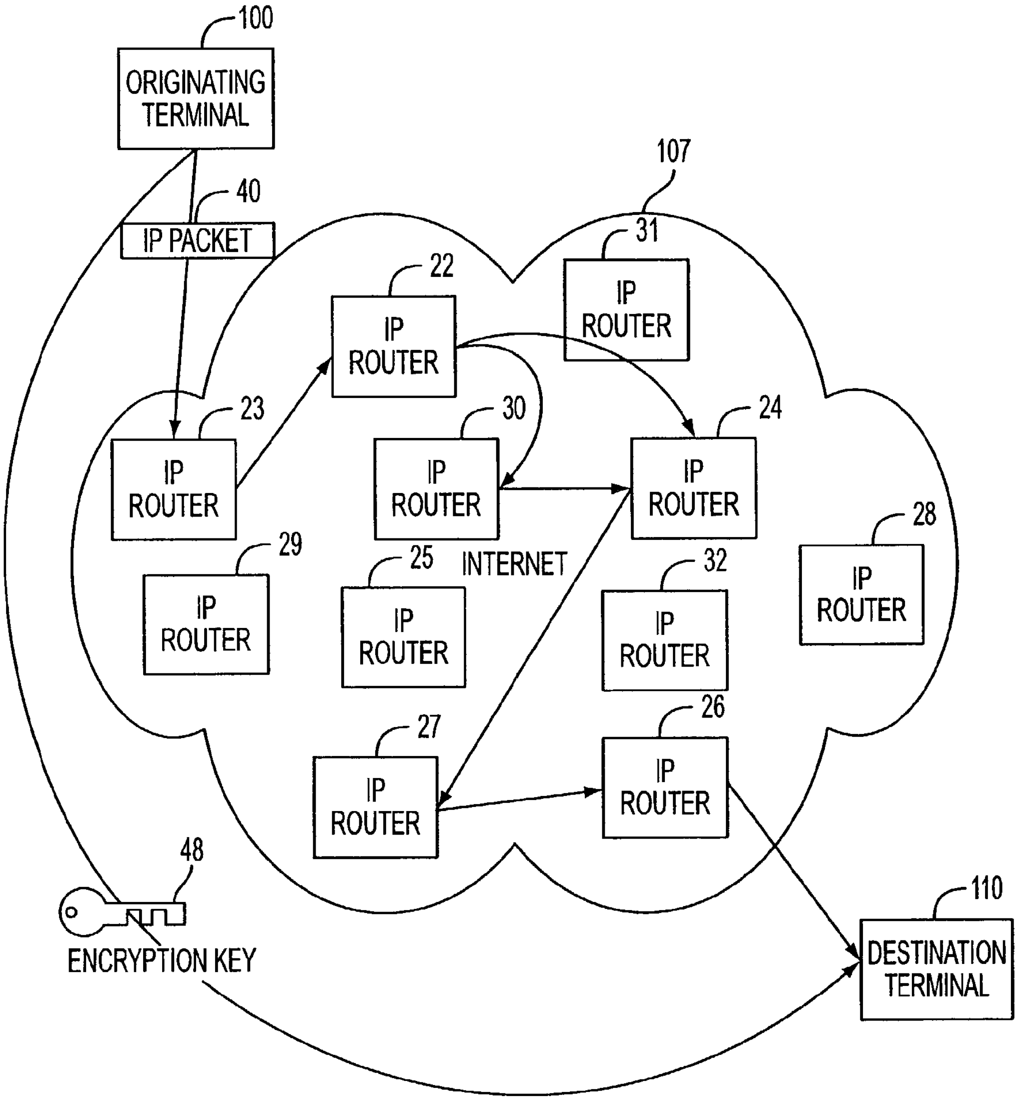

100
ORIGINATING TERMINAL
40
IP PACKET
107
31
IP ROUTER
22
IP ROUTER
23
IP ROUTER
30
IP ROUTER
24
IP ROUTER
28
IP ROUTER
29
IP ROUTER
25
IP ROUTER
INTERNET
32
IP ROUTER
26
IP ROUTER
27
IP ROUTER
48
ENCRYPTION KEY
110
DESTINATION TERMINAL

FIG. 1

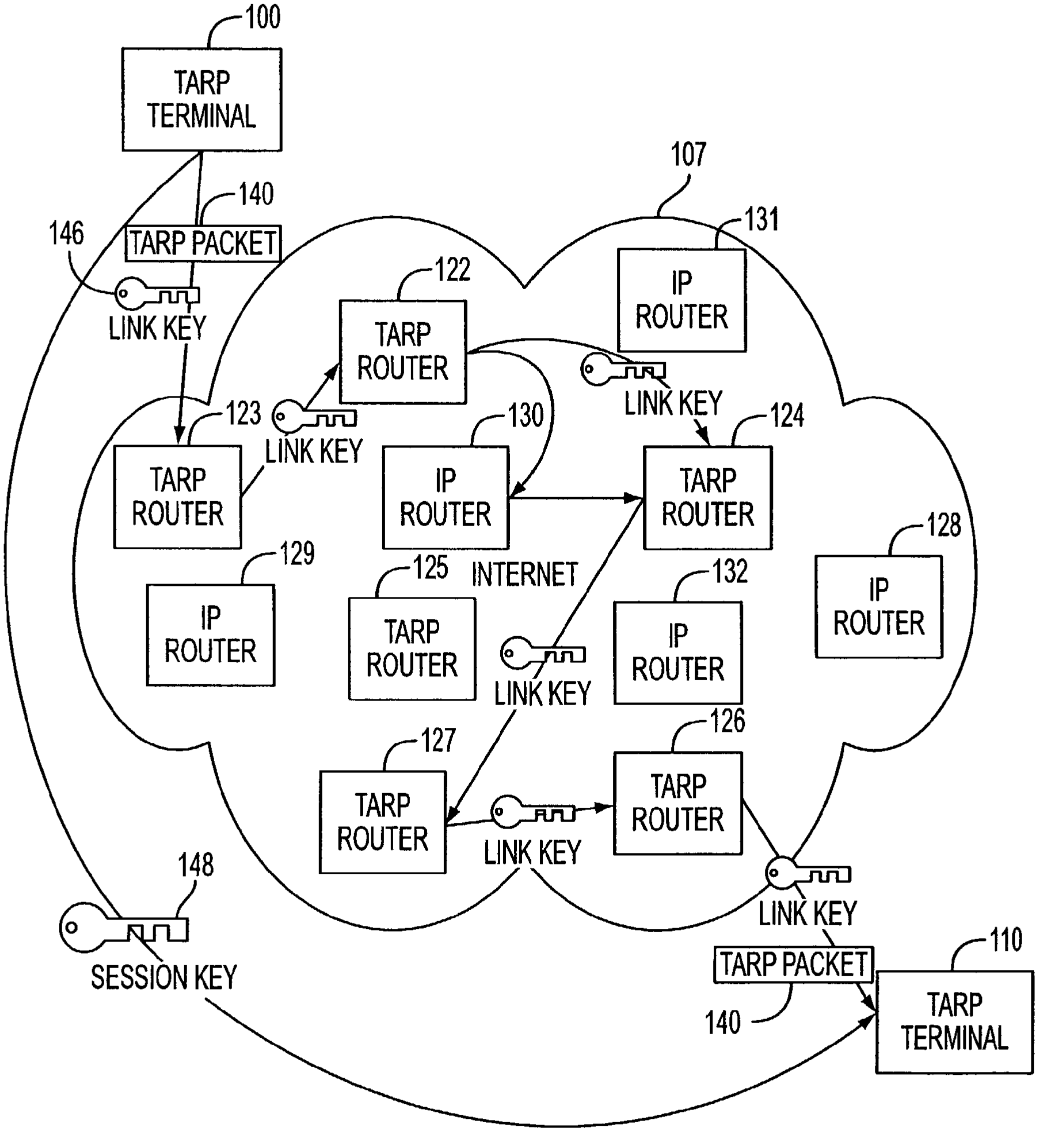

100
TARP TERMINAL
140
TARP PACKET
146
LINK KEY
107
131
IP ROUTER
122
TARP ROUTER
LINK KEY
123
TARP ROUTER
LINK KEY
130
IP ROUTER
124
TARP ROUTER
129
IP ROUTER
125
TARP ROUTER
INTERNET
132
IP ROUTER
128
IP ROUTER
LINK KEY
127
TARP ROUTER
LINK KEY
126
TARP ROUTER
LINK KEY
148
SESSION KEY
TARP PACKET
140
110
TARP TERMINAL

FIG. 2

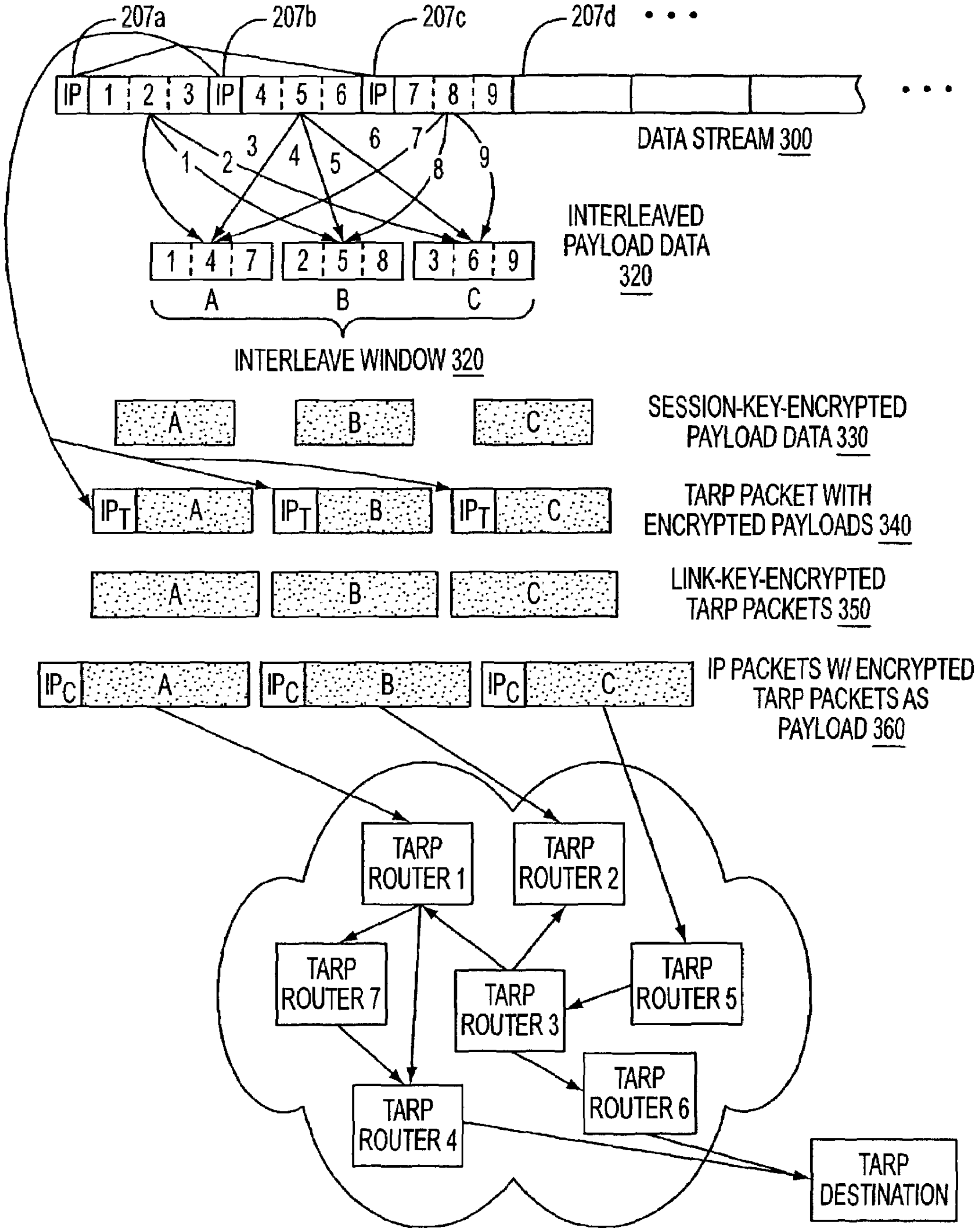

207a
207b
207c
207d
IP
1
2
3
IP
4
5
6
IP
7
8
9
DATA STREAM 300
1
2
3
4
5
6
7
8
9
INTERLEAVED PAYLOAD DATA 320
1 4 7
2 5 8
3 6 9
A
B
C
INTERLEAVE WINDOW 320
A
B
C
SESSION-KEY-ENCRYPTED PAYLOAD DATA 330
IPT A
IPT B
IPT C
TARP PACKET WITH ENCRYPTED PAYLOADS 340
A
B
C
LINK-KEY-ENCRYPTED TARP PACKETS 350
IPC A
IPC B
IPC C
IP PACKETS W/ ENCRYPTED TARP PACKETS AS PAYLOAD 360
TARP ROUTER 1
TARP ROUTER 2
TARP ROUTER 7
TARP ROUTER 3
TARP ROUTER 5
TARP ROUTER 6
TARP ROUTER 4
TARP DESTINATION

FIG. 3A

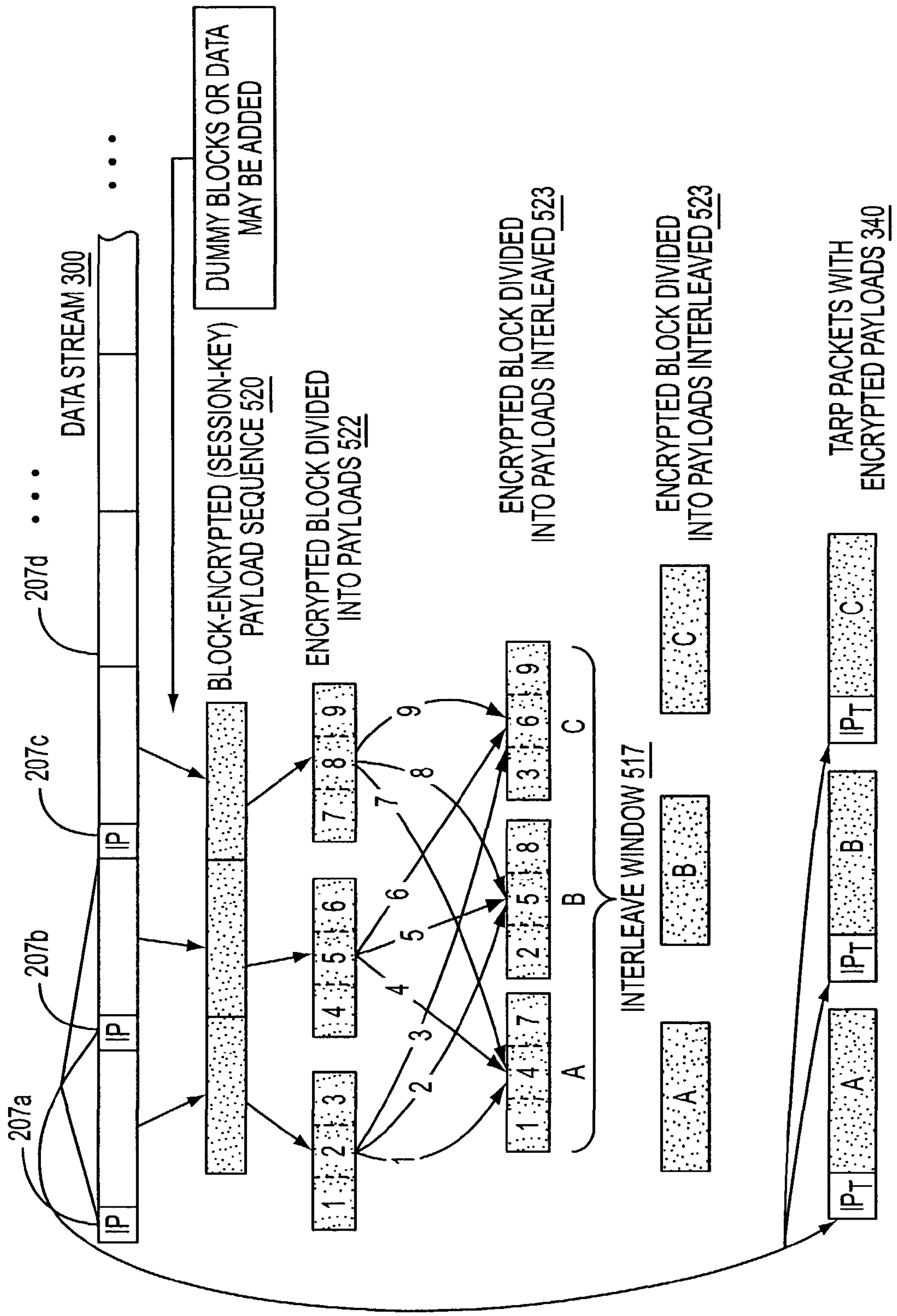

207a
207b
207c
207d
DATA STREAM 300
IP
IP
IP
DUMMY BLOCKS OR DATA MAY BE ADDED
BLOCK-ENCRYPTED (SESSION-KEY) PAYLOAD SEQUENCE 520
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS 522
1 2 3
4 5 6
7 8 9
1
2
3
4
5
6
7
8
9
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED 523
1 4 7
2 5 8
3 6 9
A
B
C
INTERLEAVE WINDOW 517
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED 523
A
B
C
TARP PACKETS WITH ENCRYPTED PAYLOADS 340
IP_T A
IP_T B
IP_T C

FIG. 3B

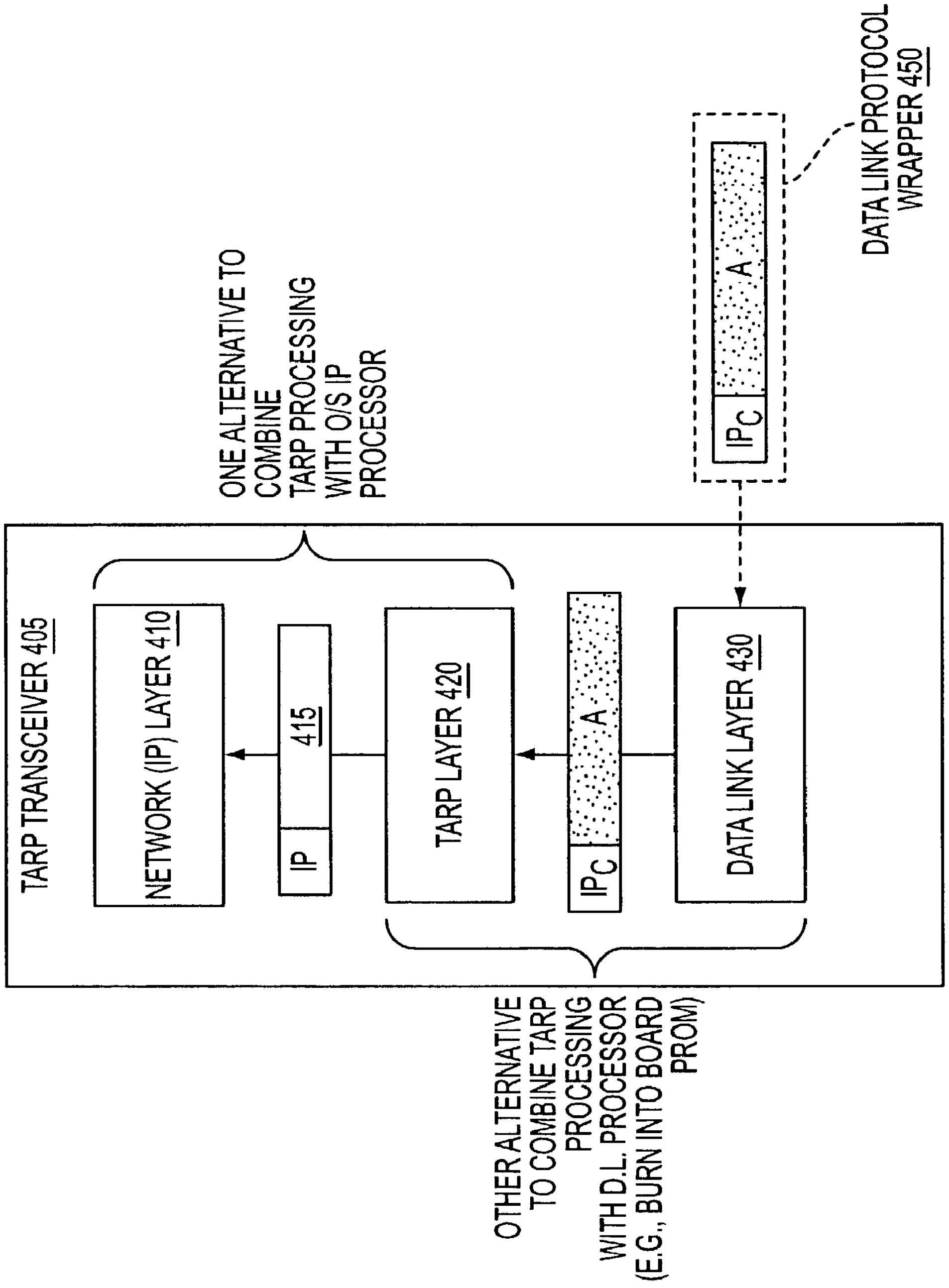

TARP TRANSCEIVER 405
NETWORK (IP) LAYER 410
IP
415
TARP LAYER 420
IP_C
A
DATA LINK LAYER 430
ONE ALTERNATIVE TO COMBINE TARP PROCESSING WITH O/S IP PROCESSOR
OTHER ALTERNATIVE TO COMBINE TARP PROCESSING WITH D.L. PROCESSOR (E.G., BURN INTO BOARD PROM)
IP_C
A
DATA LINK PROTOCOL WRAPPER 450

FIG. 4

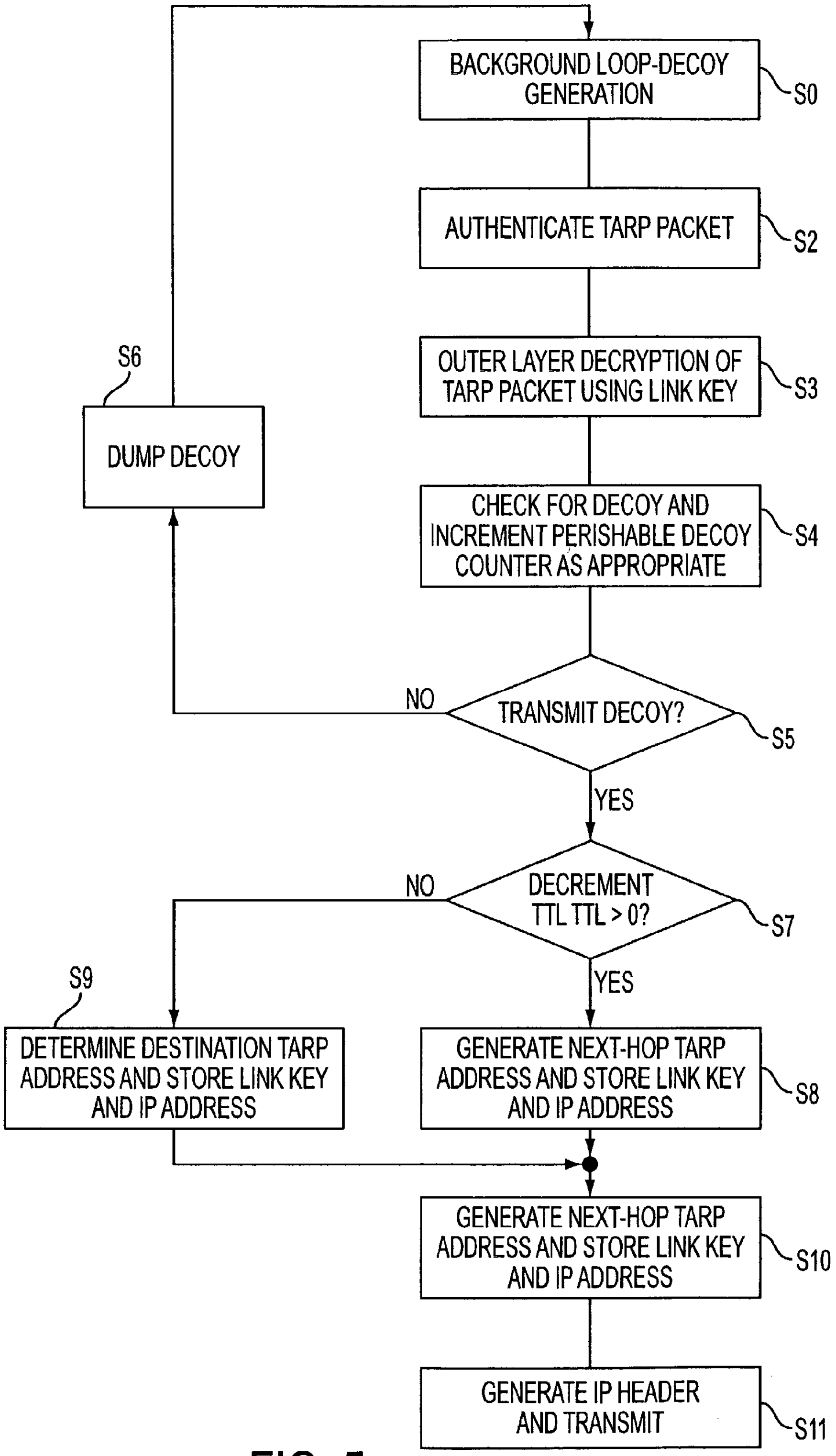

BACKGROUND LOOP-DECOY GENERATION
S0
AUTHENTICATE TARP PACKET
S2
OUTER LAYER DECRYPTION OF TARP PACKET USING LINK KEY
S3
S6
DUMP DECOY
CHECK FOR DECOY AND INCREMENT PERISHABLE DECOY COUNTER AS APPROPRIATE
S4
NO
TRANSMIT DECOY?
S5
YES
NO
DECREMENT TTL TTL > 0?
S7
YES
S9
DETERMINE DESTINATION TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S8
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S10
GENERATE IP HEADER AND TRANSMIT
S11

FIG. 5

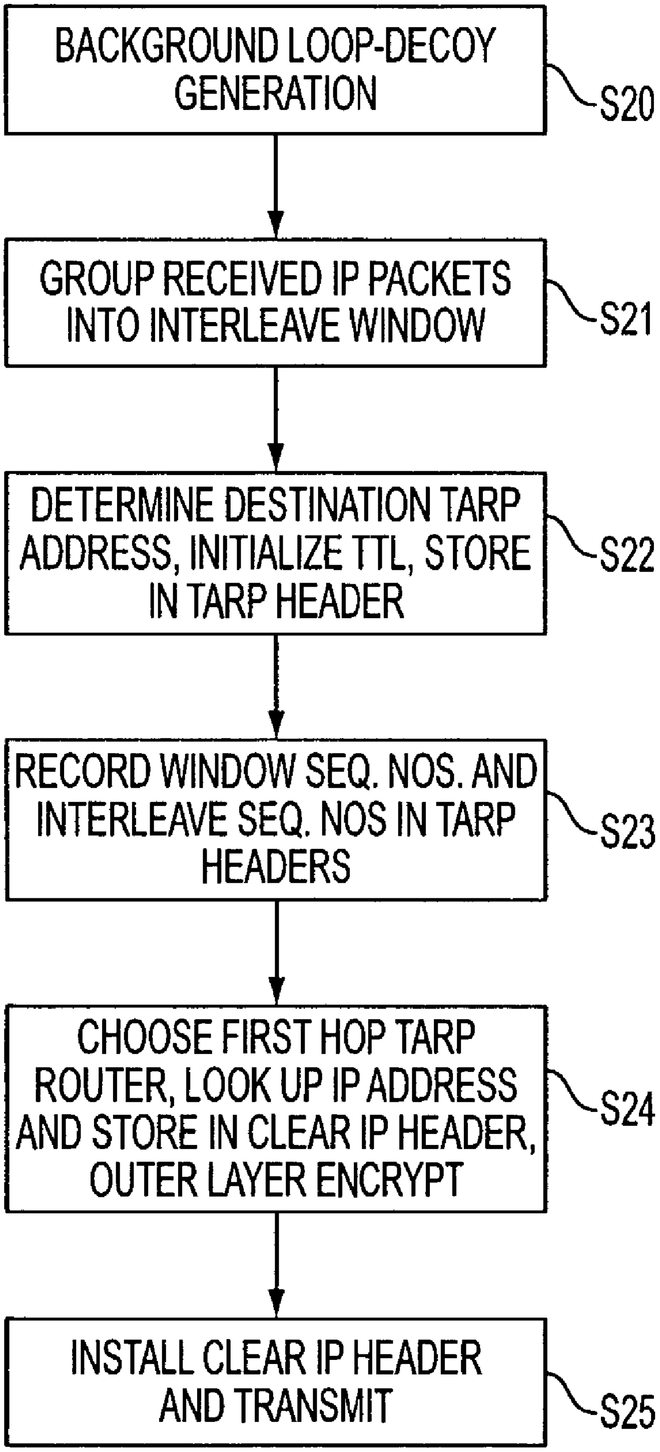

BACKGROUND LOOP-DECOY GENERATION
S20
GROUP RECEIVED IP PACKETS INTO INTERLEAVE WINDOW
S21
DETERMINE DESTINATION TARP ADDRESS, INITIALIZE TTL, STORE IN TARP HEADER
S22
RECORD WINDOW SEQ. NOS. AND INTERLEAVE SEQ. NOS IN TARP HEADERS
S23
CHOOSE FIRST HOP TARP ROUTER, LOOK UP IP ADDRESS AND STORE IN CLEAR IP HEADER, OUTER LAYER ENCRYPT
S24
INSTALL CLEAR IP HEADER AND TRANSMIT
S25

FIG. 6

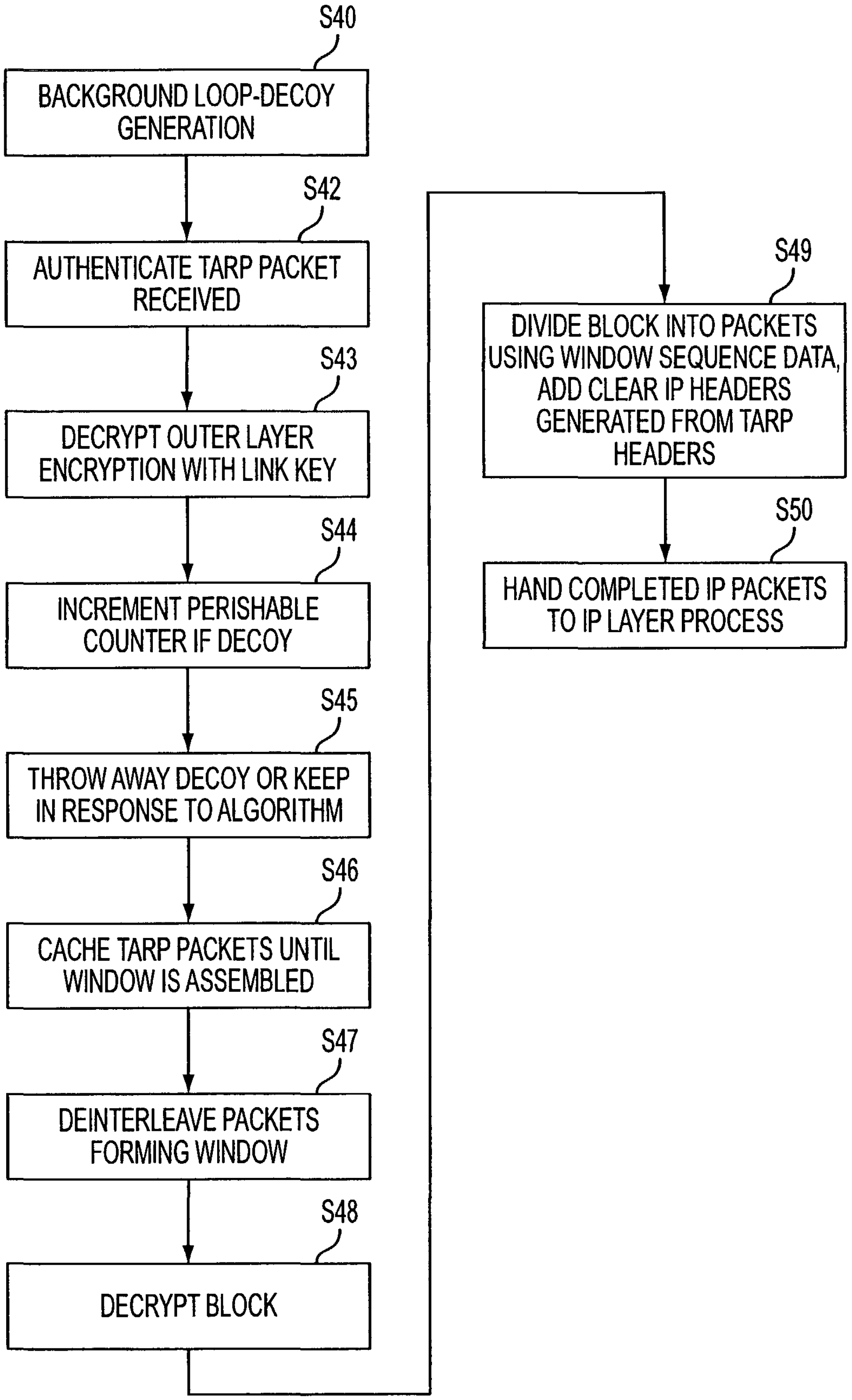

S40
BACKGROUND LOOP-DECOY GENERATION
S42
AUTHENTICATE TARP PACKET RECEIVED
S43
DECRYPT OUTER LAYER ENCRYPTION WITH LINK KEY
S44
INCREMENT PERISHABLE COUNTER IF DECOY
S45
THROW AWAY DECOY OR KEEP IN RESPONSE TO ALGORITHM
S46
CACHE TARP PACKETS UNTIL WINDOW IS ASSEMBLED
S47
DEINTERLEAVE PACKETS FORMING WINDOW
S48
DECRYPT BLOCK
S49
DIVIDE BLOCK INTO PACKETS USING WINDOW SEQUENCE DATA, ADD CLEAR IP HEADERS GENERATED FROM TARP HEADERS
S50
HAND COMPLETED IP PACKETS TO IP LAYER PROCESS

FIG. 7

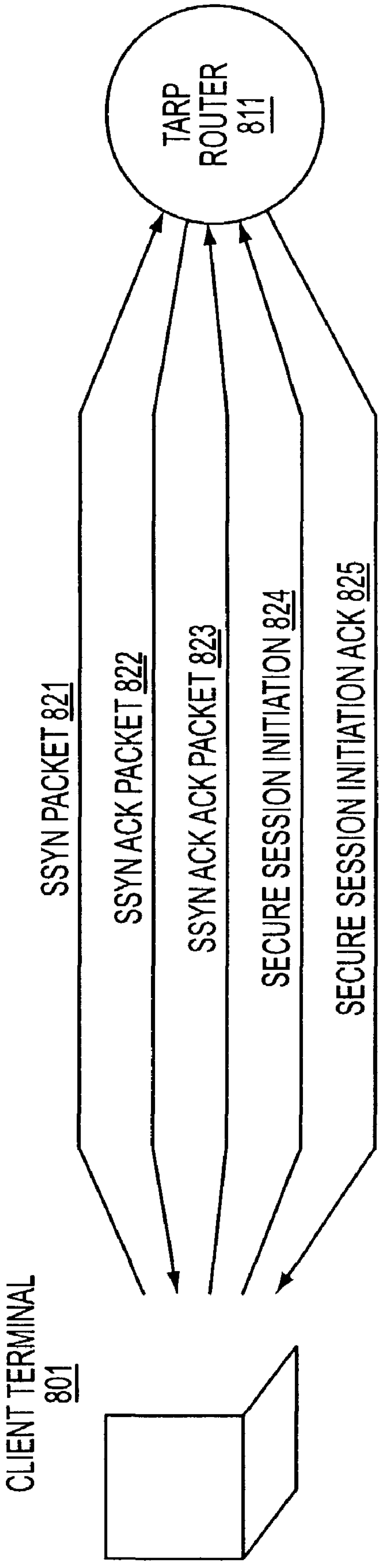
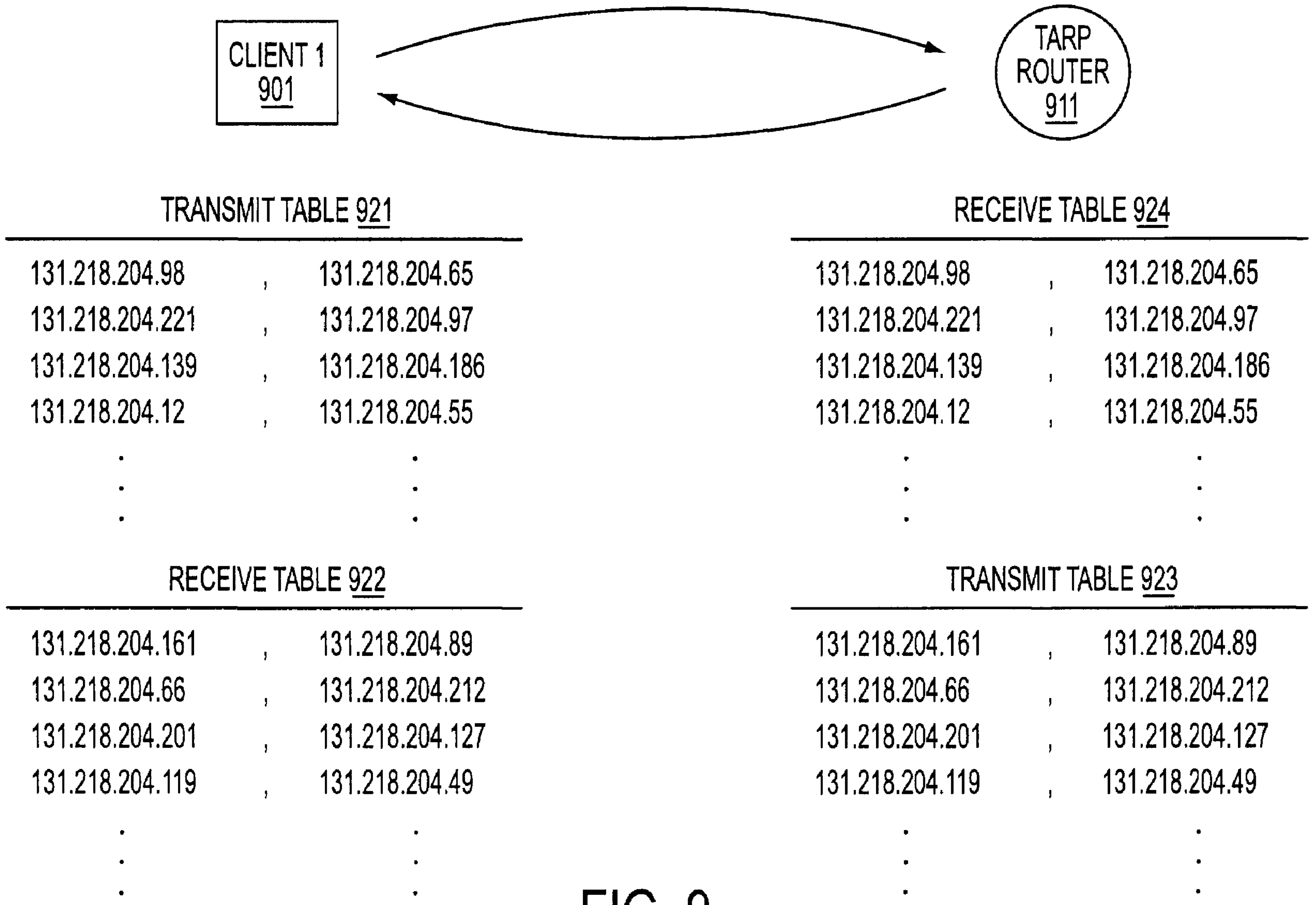

TARP ROUTER 811
SSYN PACKET 821
SSYN ACK PACKET 822
SSYN ACK ACK PACKET 823
SECURE SESSION INITIATION 824
SECURE SESSION INITIATION ACK 825
CLIENT TERMINAL 801

FIG. 8



TRANSMIT TABLE 921

131.218.204.98	,	131.218.204.65
131.218.204.221	,	131.218.204.97
131.218.204.139	,	131.218.204.186
131.218.204.12	,	131.218.204.55
⋮		⋮

RECEIVE TABLE 924

131.218.204.98	,	131.218.204.65
131.218.204.221	,	131.218.204.97
131.218.204.139	,	131.218.204.186
131.218.204.12	,	131.218.204.55
⋮		⋮

RECEIVE TABLE 922

131.218.204.161	,	131.218.204.89
131.218.204.66	,	131.218.204.212
131.218.204.201	,	131.218.204.127
131.218.204.119	,	131.218.204.49
⋮		⋮

TRANSMIT TABLE 923

131.218.204.161	,	131.218.204.89
131.218.204.66	,	131.218.204.212
131.218.204.201	,	131.218.204.127
131.218.204.119	,	131.218.204.49
⋮		⋮

FIG. 9

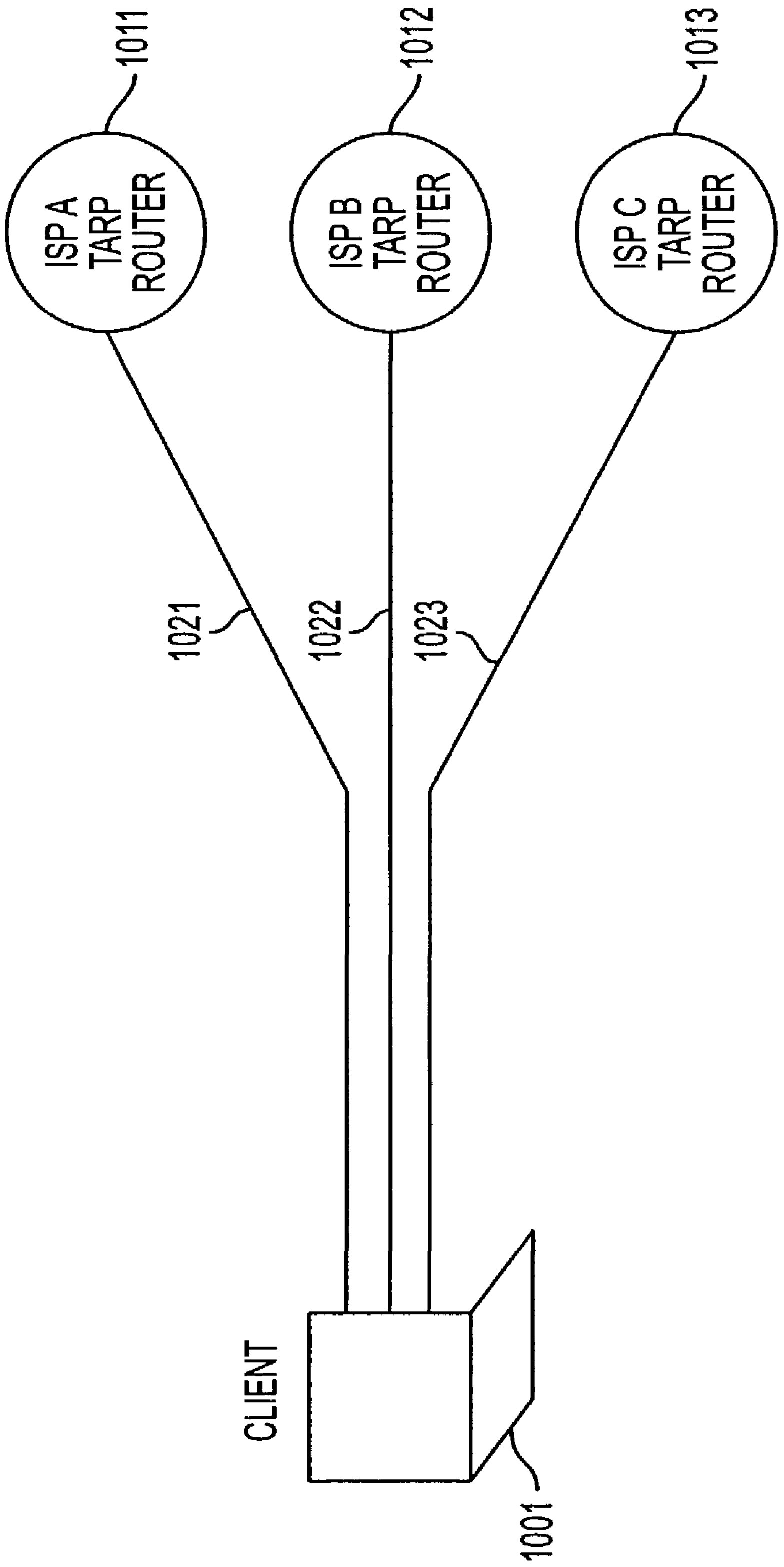

1011
1012
1013
ISP A TARP ROUTER
ISP B TARP ROUTER
ISP C TARP ROUTER
1021
1022
1023
CLIENT
1001

FIG. 10

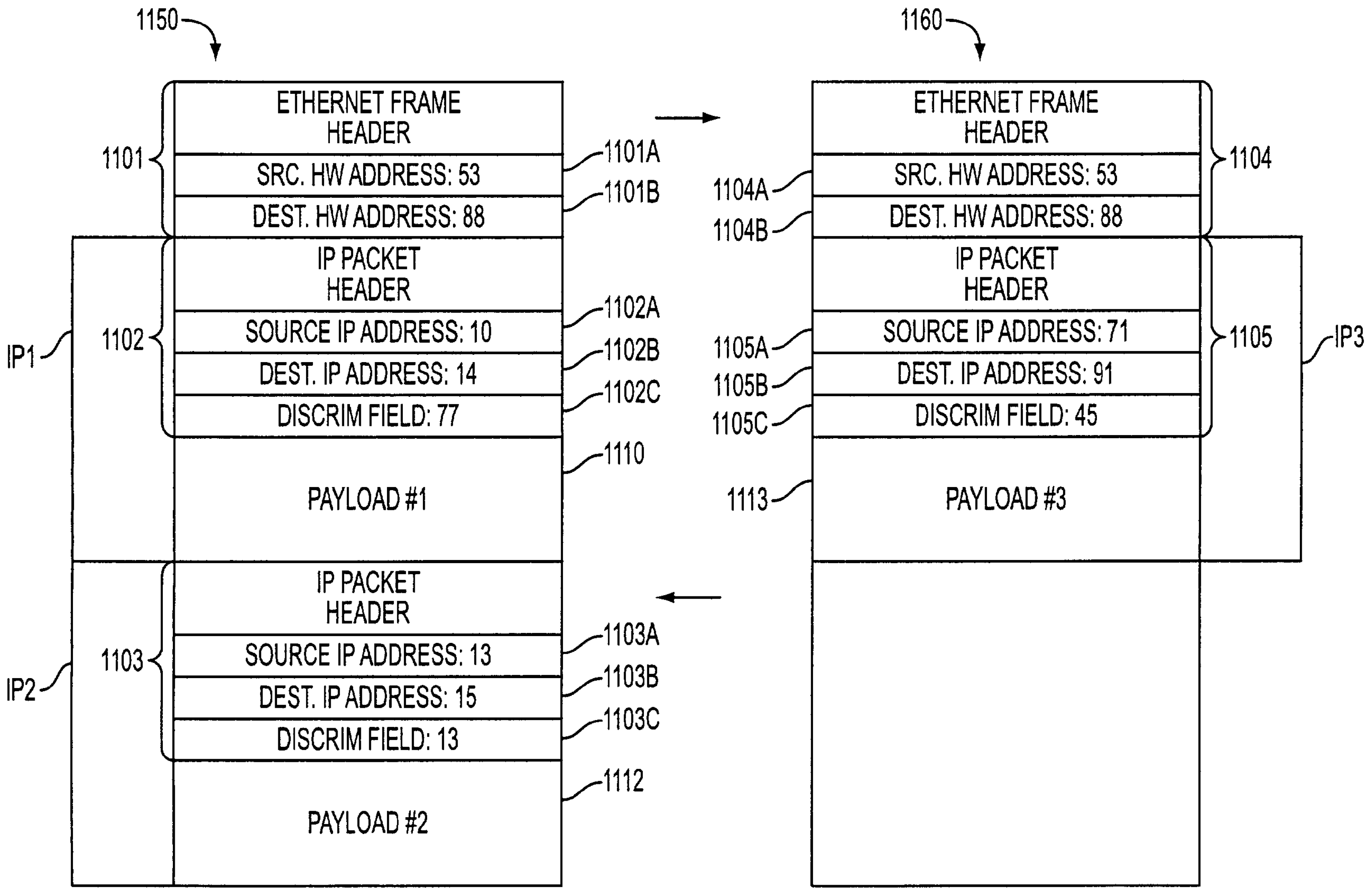
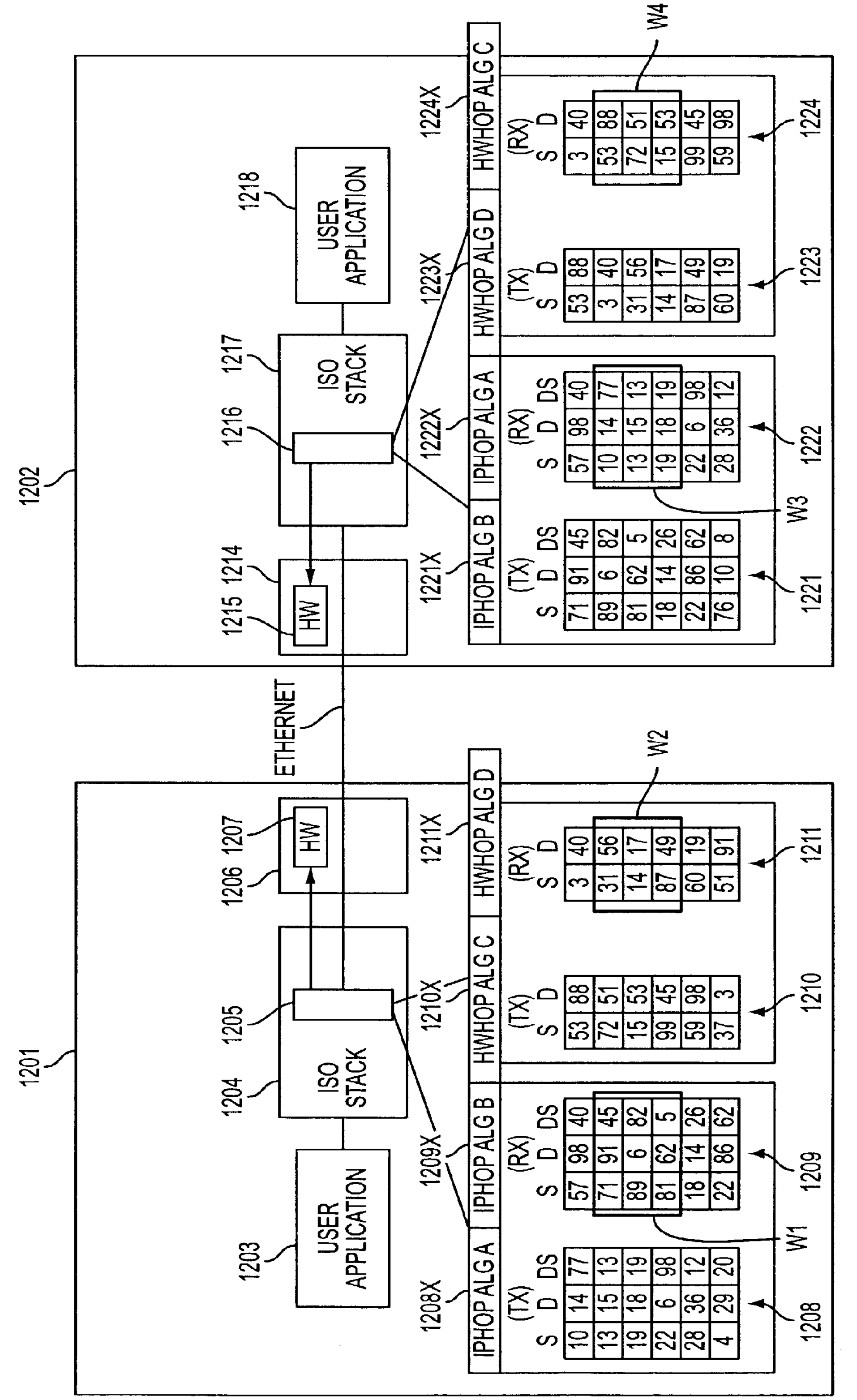

1150
ETHERNET FRAME HEADER
SRC. HW ADDRESS: 53
DEST. HW ADDRESS: 88
IP PACKET HEADER
SOURCE IP ADDRESS: 10
DEST. IP ADDRESS: 14
DISCRIM FIELD: 77
PAYLOAD #1
IP PACKET HEADER
SOURCE IP ADDRESS: 13
DEST. IP ADDRESS: 15
DISCRIM FIELD: 13
PAYLOAD #2
1101
1101A
1101B
1102
1102A
1102B
1102C
1110
1103
1103A
1103B
1103C
1112
IP1
IP2
1160
ETHERNET FRAME HEADER
SRC. HW ADDRESS: 53
DEST. HW ADDRESS: 88
IP PACKET HEADER
SOURCE IP ADDRESS: 71
DEST. IP ADDRESS: 91
DISCRIM FIELD: 45
PAYLOAD #3
1104
1104A
1104B
1105
1105A
1105B
1105C
1113
IP3

FIG. 11



1208 IPHOP ALG A (TX)

S	D	DS
10	14	77
13	15	13
19	18	19
22	6	98
28	36	12
4	29	20

1209 IPHOP ALG B (RX)

S	D	DS
57	98	40
71	91	45
89	6	82
81	62	5
18	14	26
22	86	62

1210 HWHOP ALG C (TX)

S	D
53	88
72	51
15	53
99	45
59	98
37	3

1211 HWHOP ALG D (RX)

S	D
3	40
31	56
14	17
87	49
60	19
51	91

1221 IPHOP ALG B (TX)

S	D	DS
71	91	45
89	6	82
81	62	5
18	14	26
22	86	62
76	10	8

1222 IPHOP ALG A (RX)

S	D	DS
57	98	40
10	14	77
13	15	13
19	18	19
22	6	98
28	36	12

1223 HWHOP ALG D (TX)

S	D
53	88
3	40
31	56
14	17
87	49
60	19

1224 HWHOP ALG C (RX)

S	D
3	40
53	88
72	51
15	53
99	45
59	98

FIG. 12A

MODE OR EMBODIMENT	HARDWARE ADDRESSES	IP ADDRESSES	DISCRIMINATOR FIELD VALUES
1. PROMISCUOUS	SAME FOR ALL NODES OR COMPLETELY RANDOM	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
2. PROMISCUOUS PER VPN	FIXED FOR EACH VPN	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
3. HARDWARE HOPPING	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC

FIG. 12B

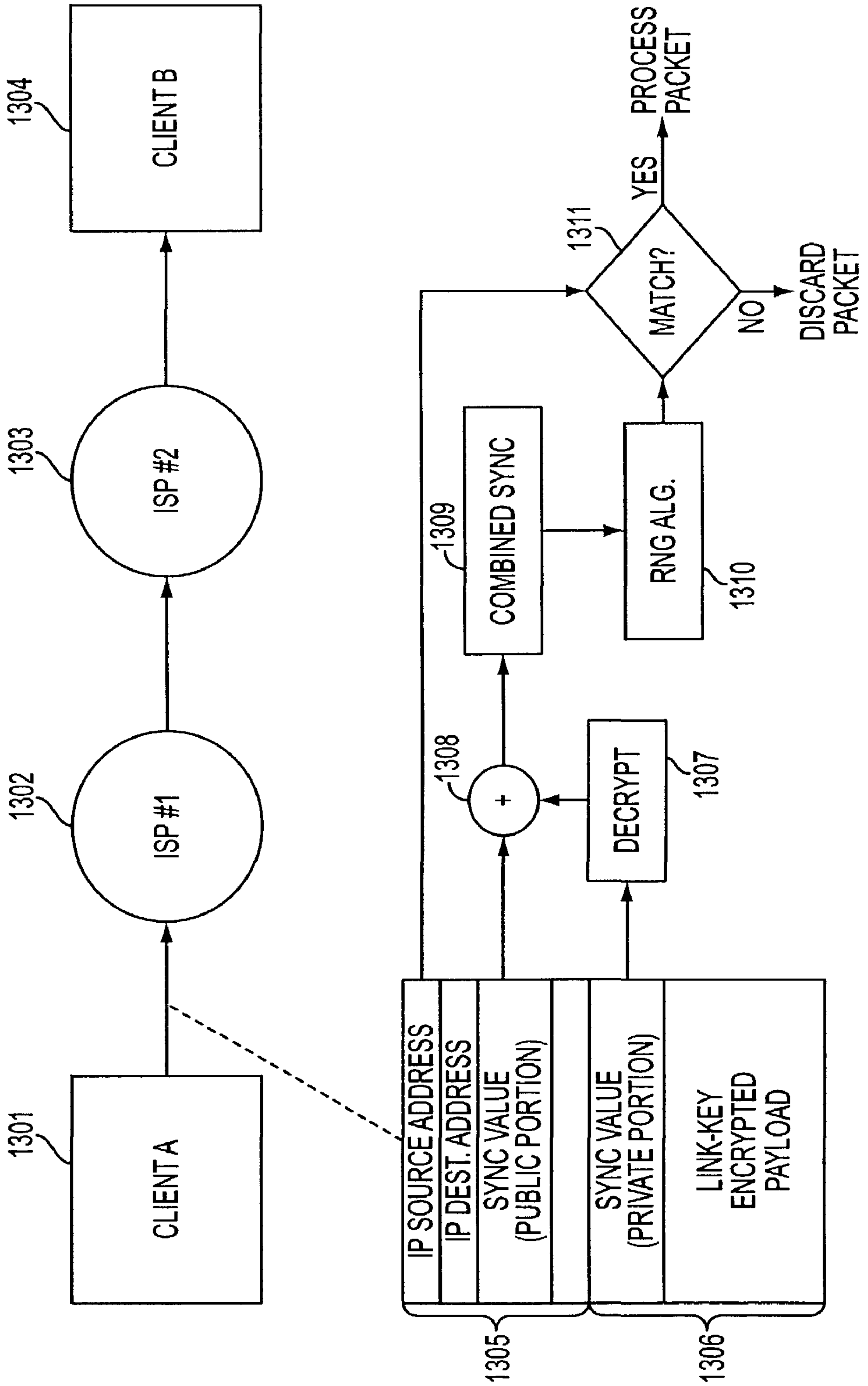

1301
CLIENT A
1302
ISP #1
1303
ISP #2
1304
CLIENT B
IP SOURCE ADDRESS
IP DEST. ADDRESS
SYNC VALUE (PUBLIC PORTION)
SYNC VALUE (PRIVATE PORTION)
LINK-KEY ENCRYPTED PAYLOAD
1305
1306
DECRYPT
1307
+
1308
COMBINED SYNC
1309
RNG ALG.
1310
MATCH?
1311
YES
PROCESS PACKET
NO
DISCARD PACKET

FIG. 13

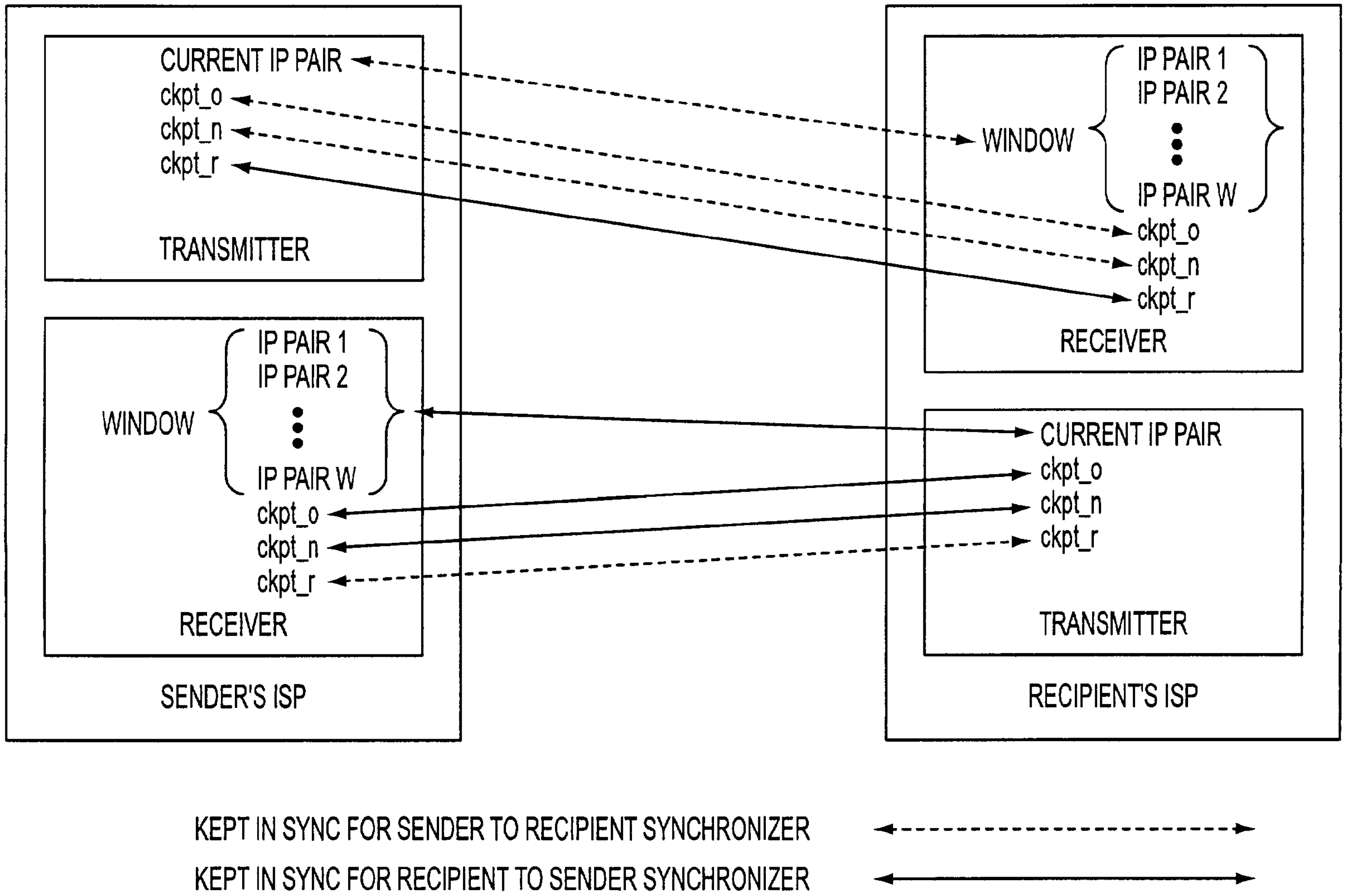
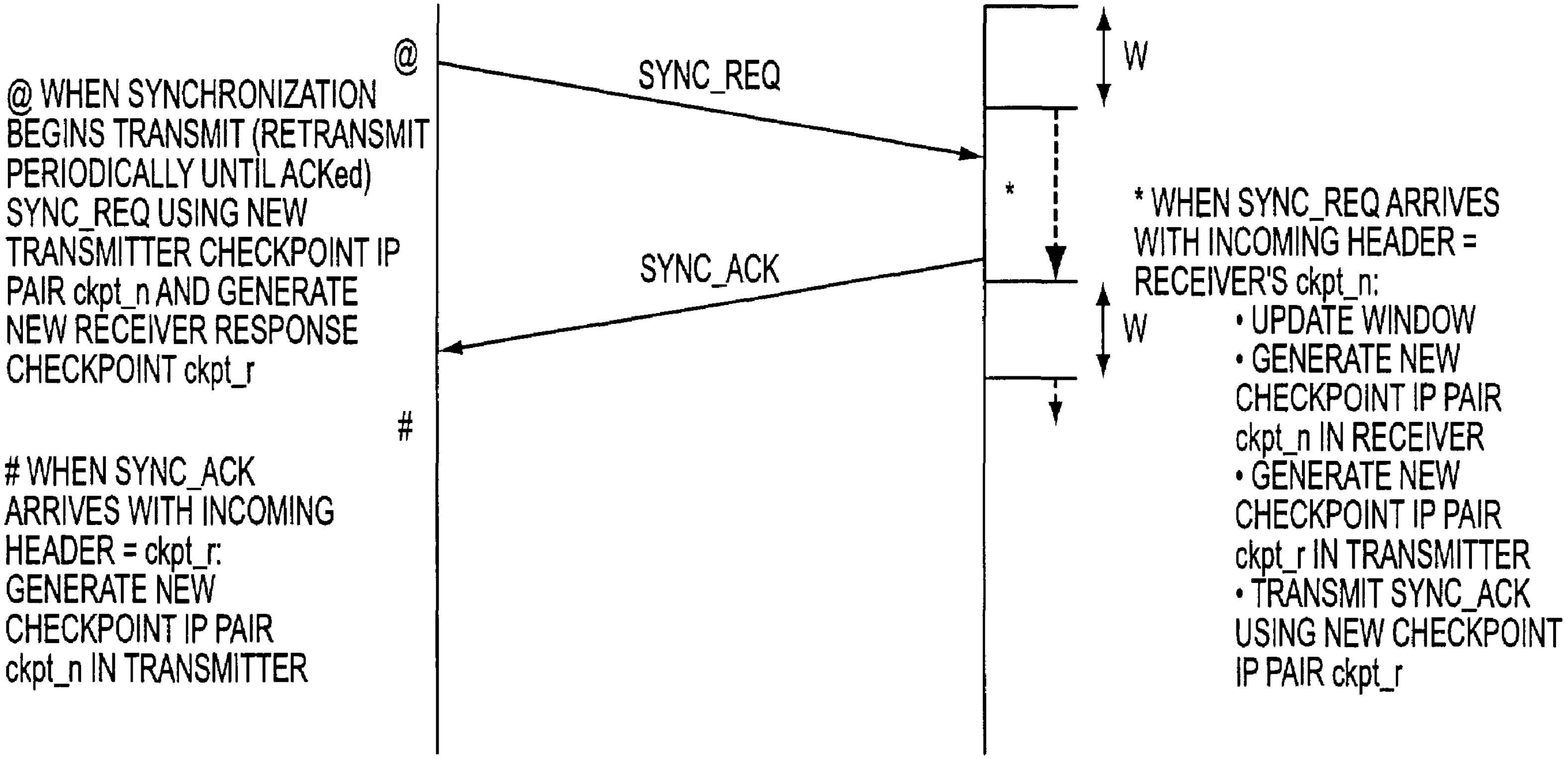

CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
SENDER'S ISP
WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
RECIPIENT'S ISP
KEPT IN SYNC FOR SENDER TO RECIPIENT SYNCHRONIZER
KEPT IN SYNC FOR RECIPIENT TO SENDER SYNCHRONIZER

FIG. 14



@ WHEN SYNCHRONIZATION BEGINS TRANSMIT (RETRANSMIT PERIODICALLY UNTIL ACKed) SYNC_REQ USING NEW TRANSMITTER CHECKPOINT IP PAIR ckpt_n AND GENERATE NEW RECEIVER RESPONSE CHECKPOINT ckpt_r

# WHEN SYNC_ACK ARRIVES WITH INCOMING HEADER = ckpt_r: GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN TRANSMITTER

* WHEN SYNC_REQ ARRIVES WITH INCOMING HEADER = RECEIVER'S ckpt_n:

- UPDATE WINDOW
- GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN RECEIVER
- GENERATE NEW CHECKPOINT IP PAIR ckpt_r IN TRANSMITTER
- TRANSMIT SYNC_ACK USING NEW CHECKPOINT IP PAIR ckpt_r

FIG. 15

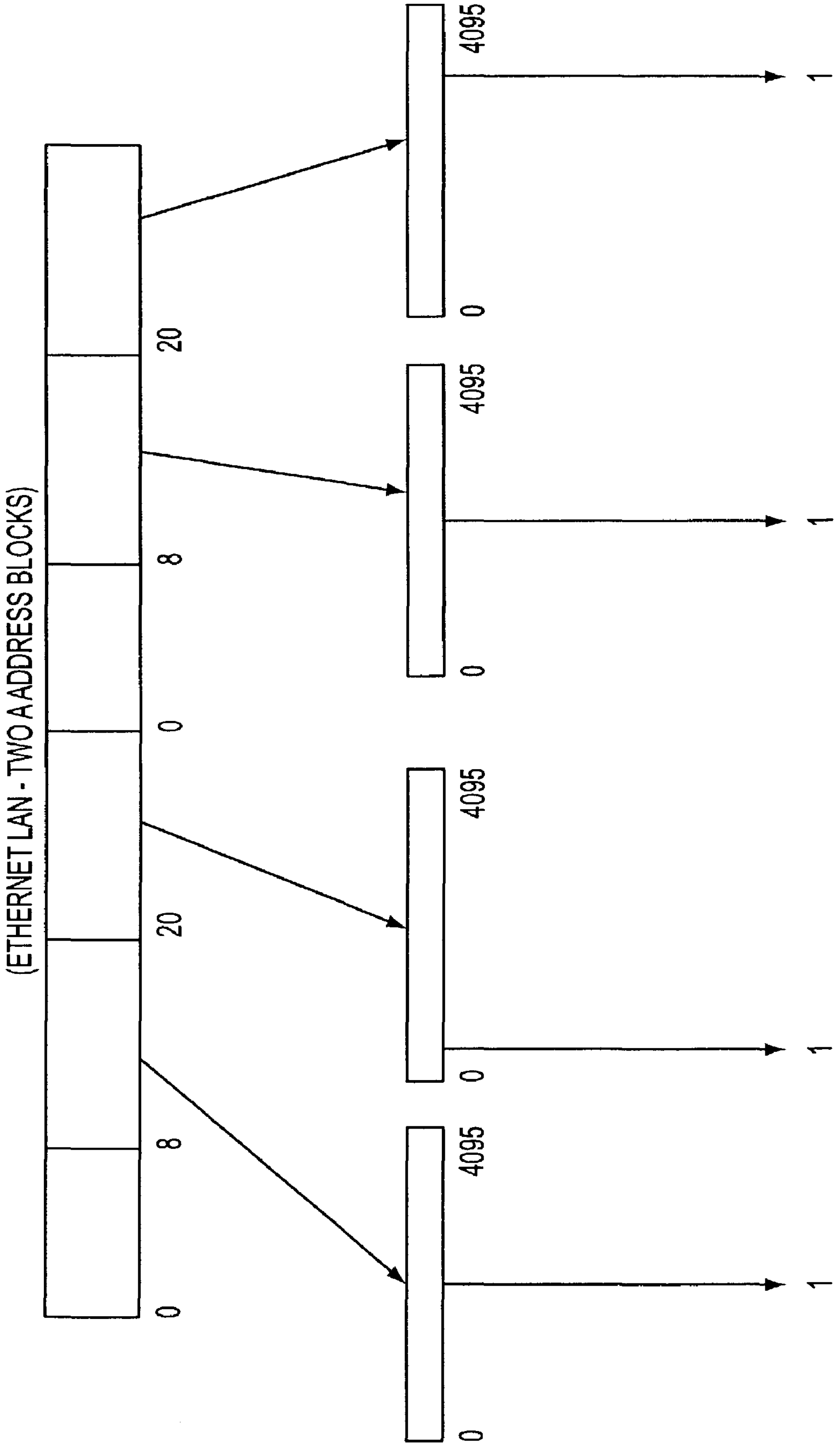

(ETHERNET LAN - TWO A ADDRESS BLOCKS)
0
8
20
0
8
20
0
4095
1
0
4095
1
0
4095
1
0
4095
1

FIG. 16

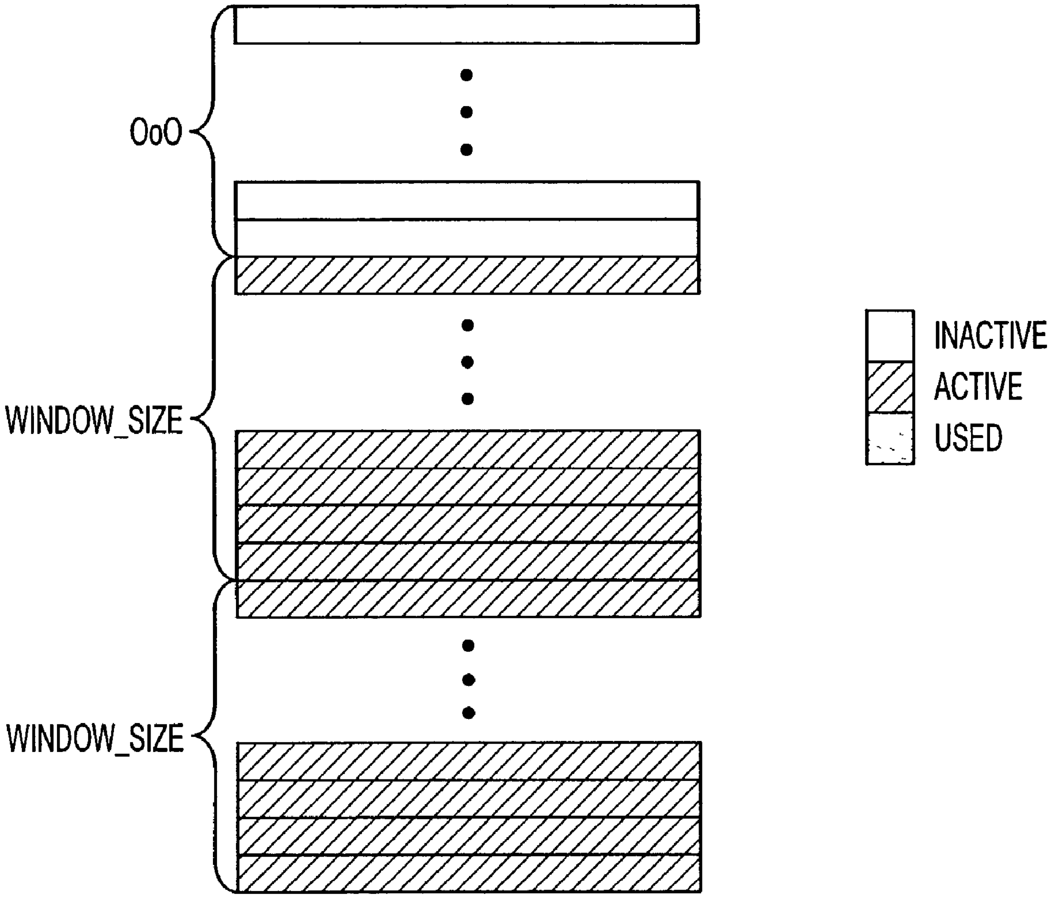


FIG. 17

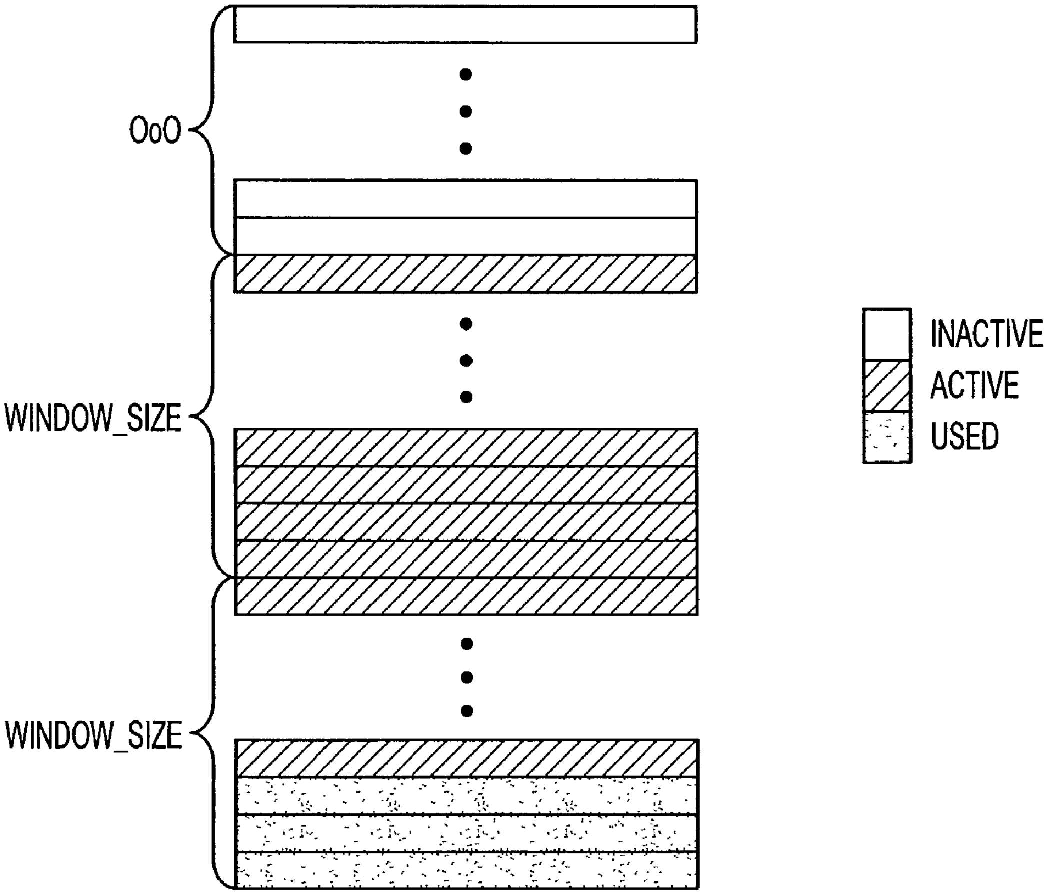


FIG. 18

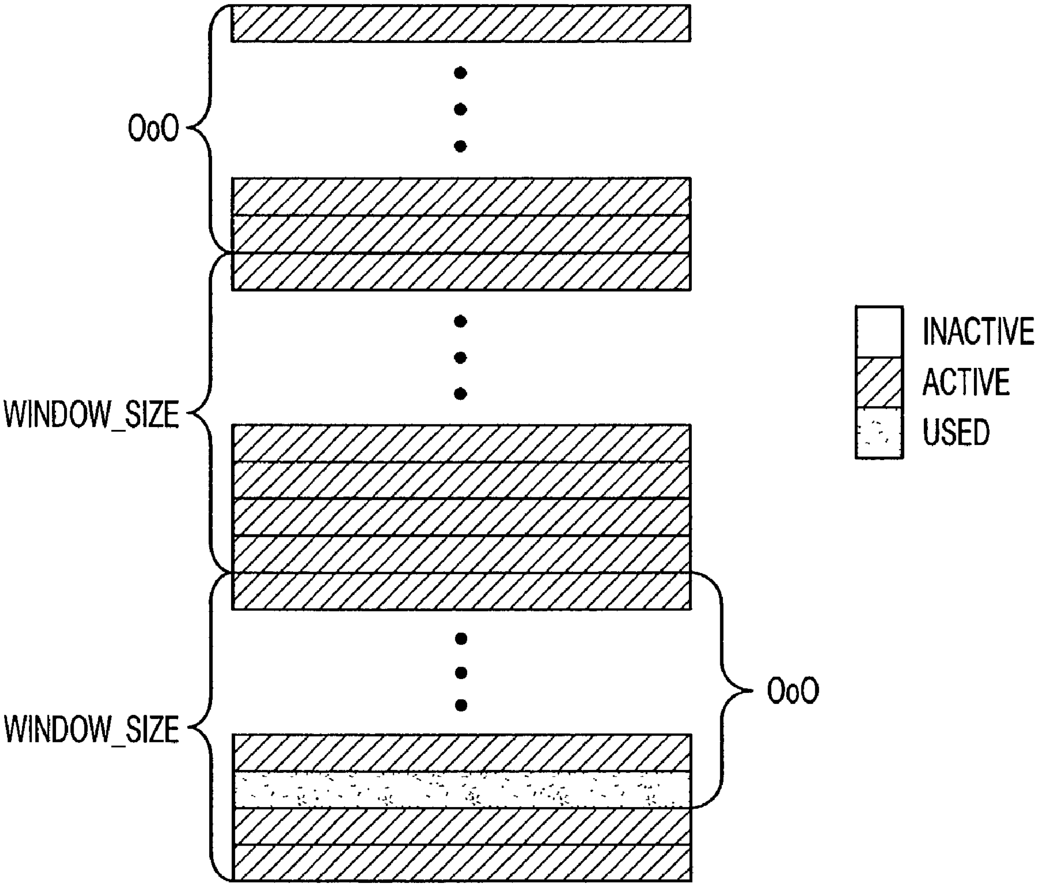


FIG. 19

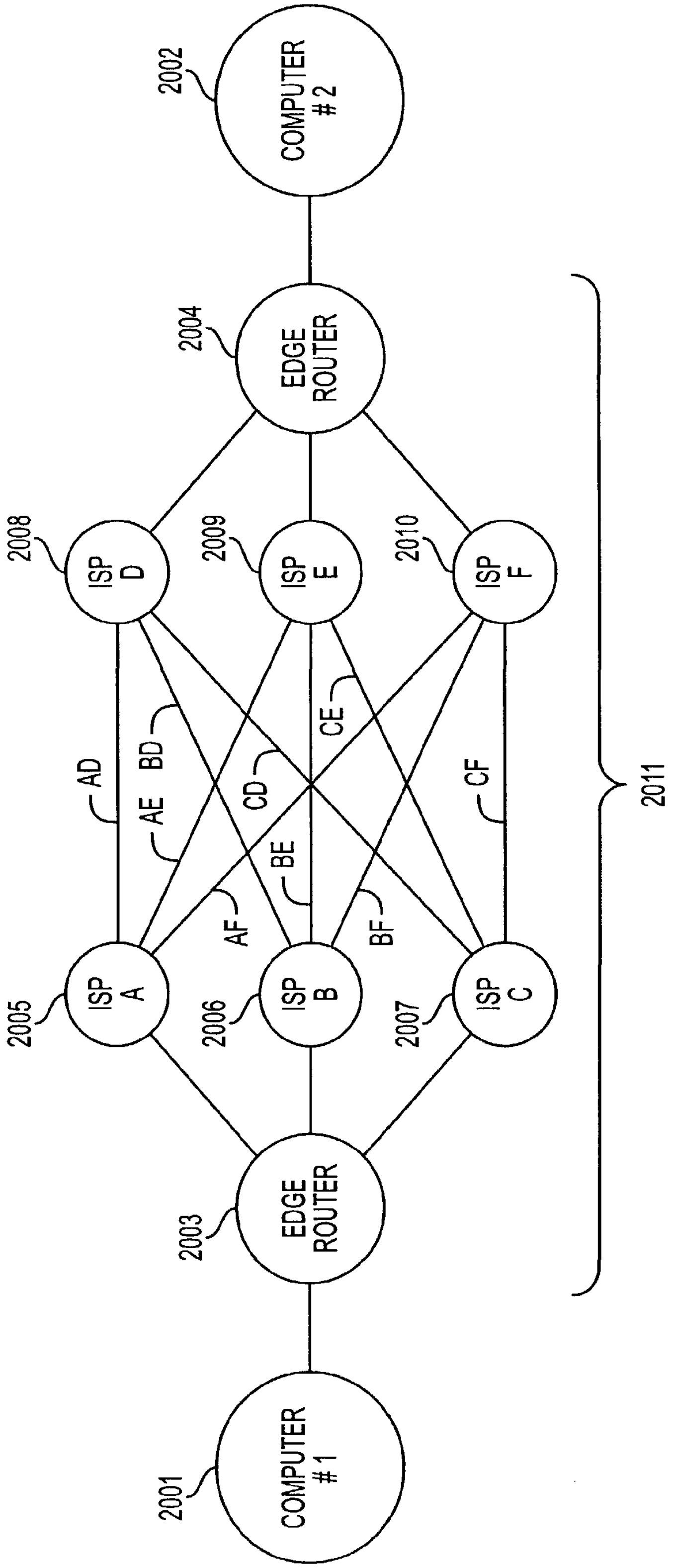

2002
COMPUTER #2
2004
EDGE ROUTER
2008
ISP D
2009
ISP E
2010
ISP F
AD
BD
CE
CD
AE
CF
BE
AF
BF
2005
ISP A
2006
ISP B
2007
ISP C
2003
EDGE ROUTER
2001
COMPUTER #1
2011

FIG. 20

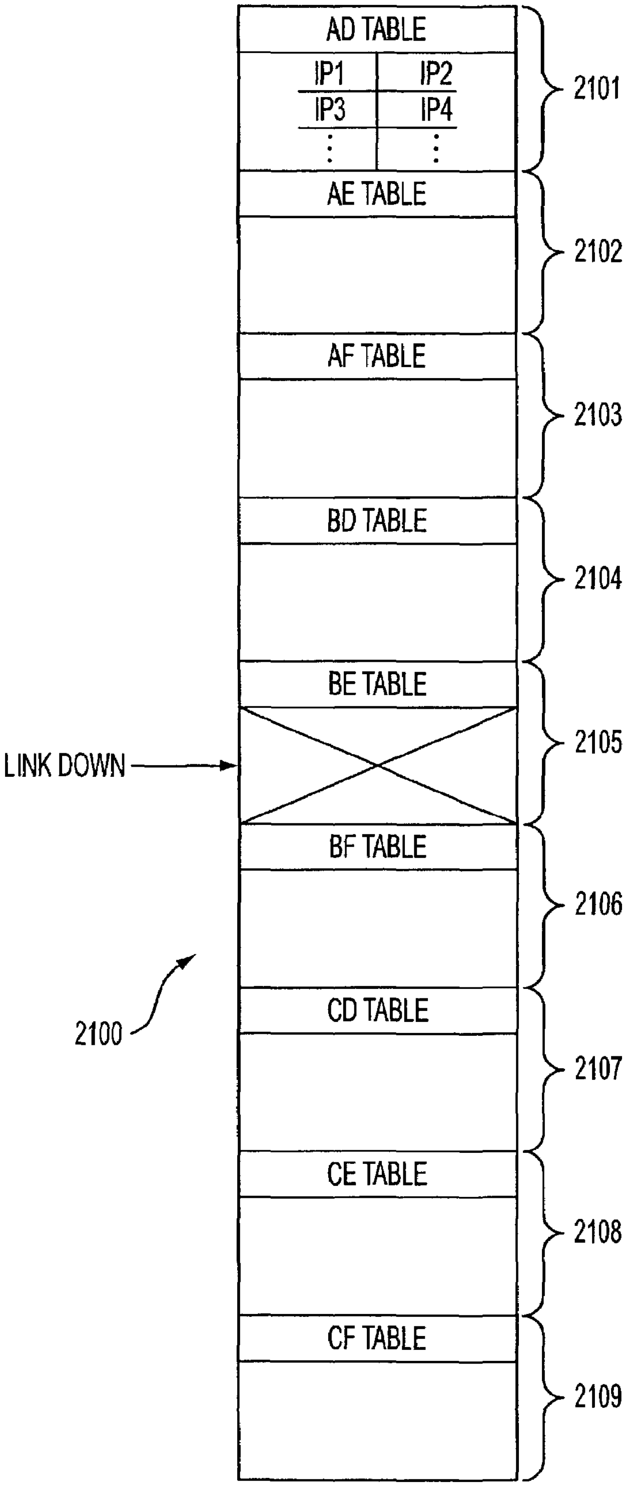

AD TABLE
IP1
IP2
IP3
IP4
2101
AE TABLE
2102
AF TABLE
2103
BD TABLE
2104
BE TABLE
LINK DOWN
2105
BF TABLE
2106
2100
CD TABLE
2107
CE TABLE
2108
CF TABLE
2109

FIG. 21

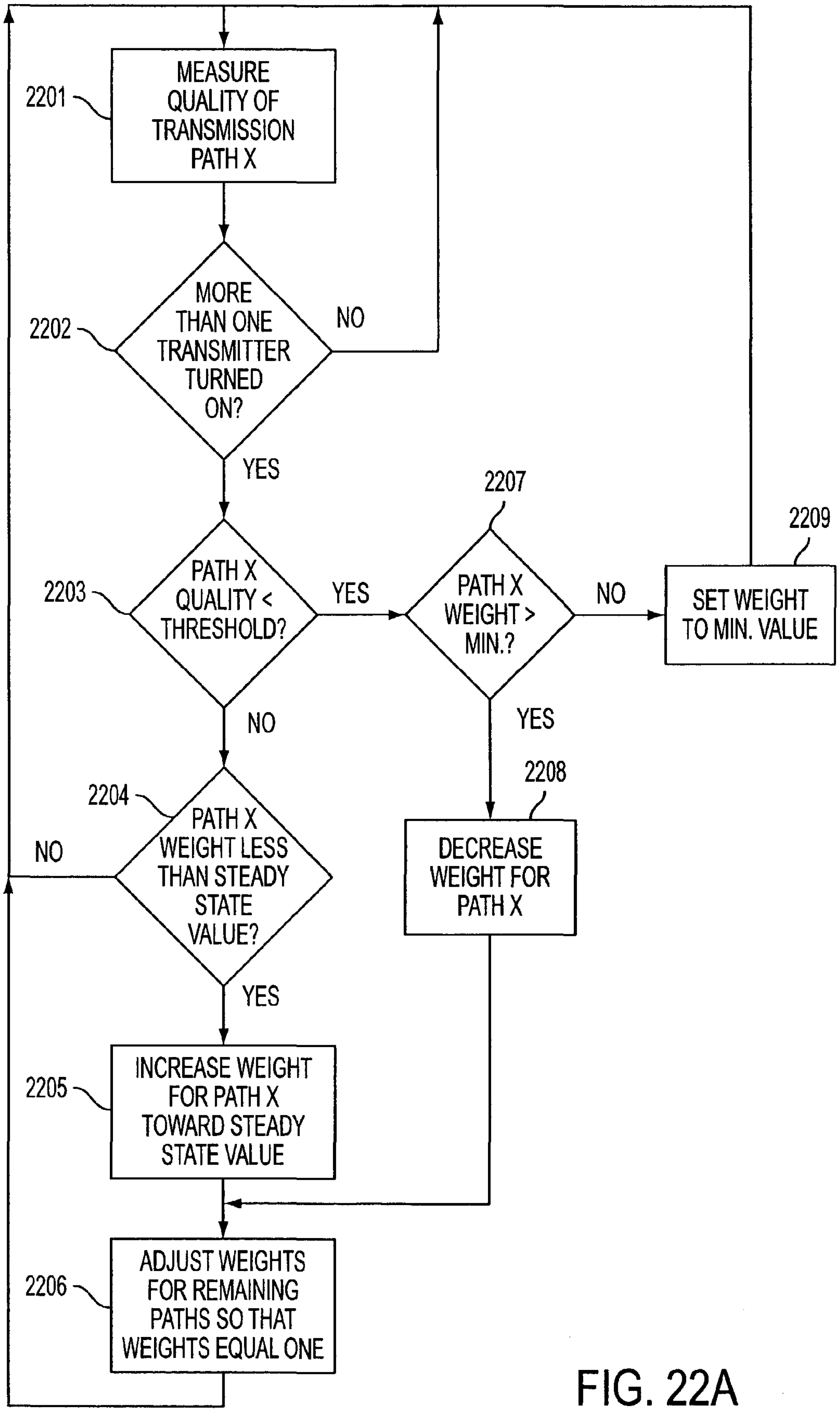

2201
MEASURE QUALITY OF TRANSMISSION PATH X
2202
MORE THAN ONE TRANSMITTER TURNED ON?
NO
YES
2203
PATH X QUALITY < THRESHOLD?
YES
NO
2207
PATH X WEIGHT > MIN.?
NO
YES
2209
SET WEIGHT TO MIN. VALUE
2208
DECREASE WEIGHT FOR PATH X
2204
PATH X WEIGHT LESS THAN STEADY STATE VALUE?
NO
YES
2205
INCREASE WEIGHT FOR PATH X TOWARD STEADY STATE VALUE
2206
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE

FIG. 22A

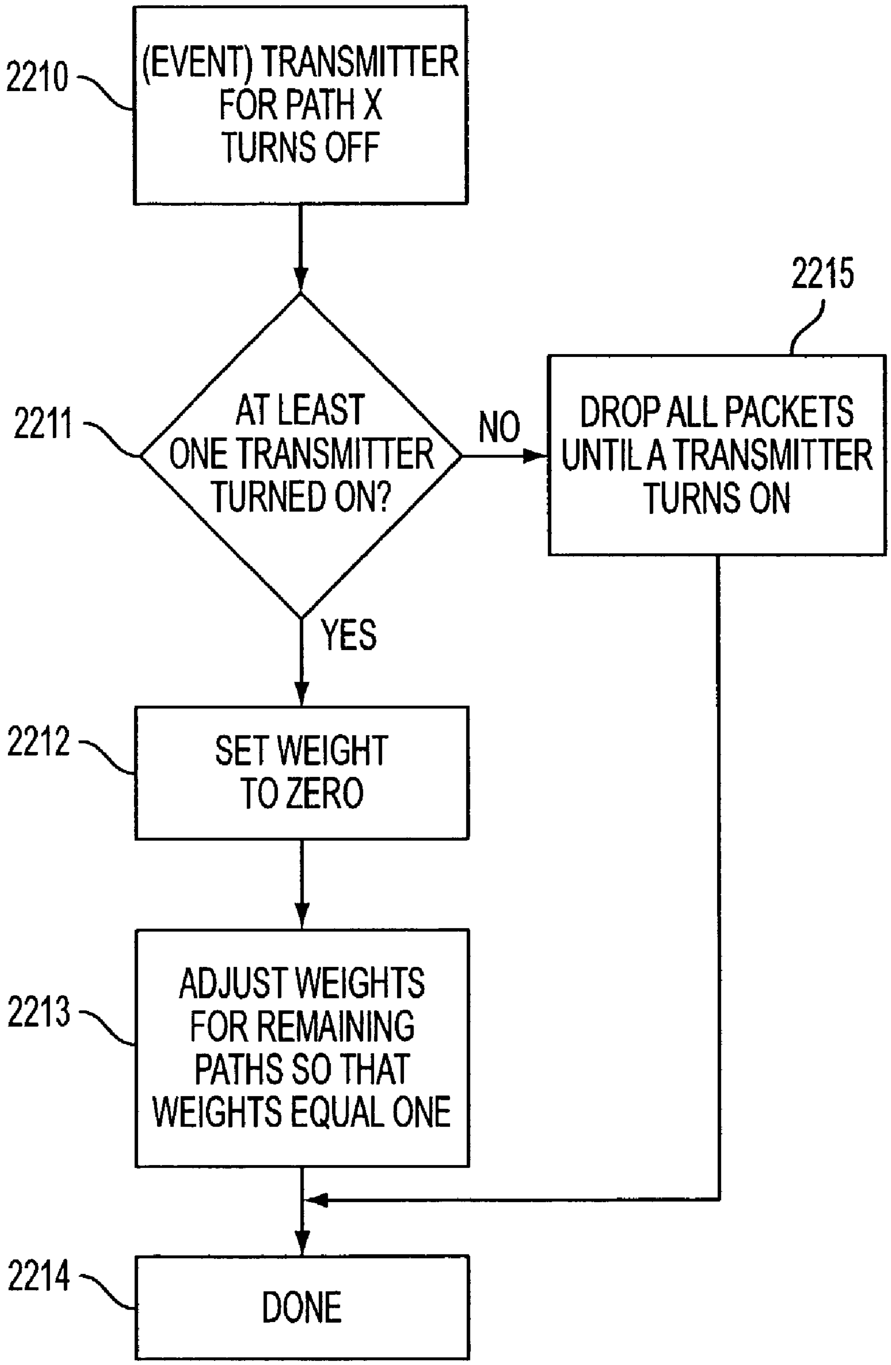

2210
(EVENT) TRANSMITTER FOR PATH X TURNS OFF
2211
AT LEAST ONE TRANSMITTER TURNED ON?
NO
2215
DROP ALL PACKETS UNTIL A TRANSMITTER TURNS ON
YES
2212
SET WEIGHT TO ZERO
2213
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE
2214
DONE

FIG. 22B

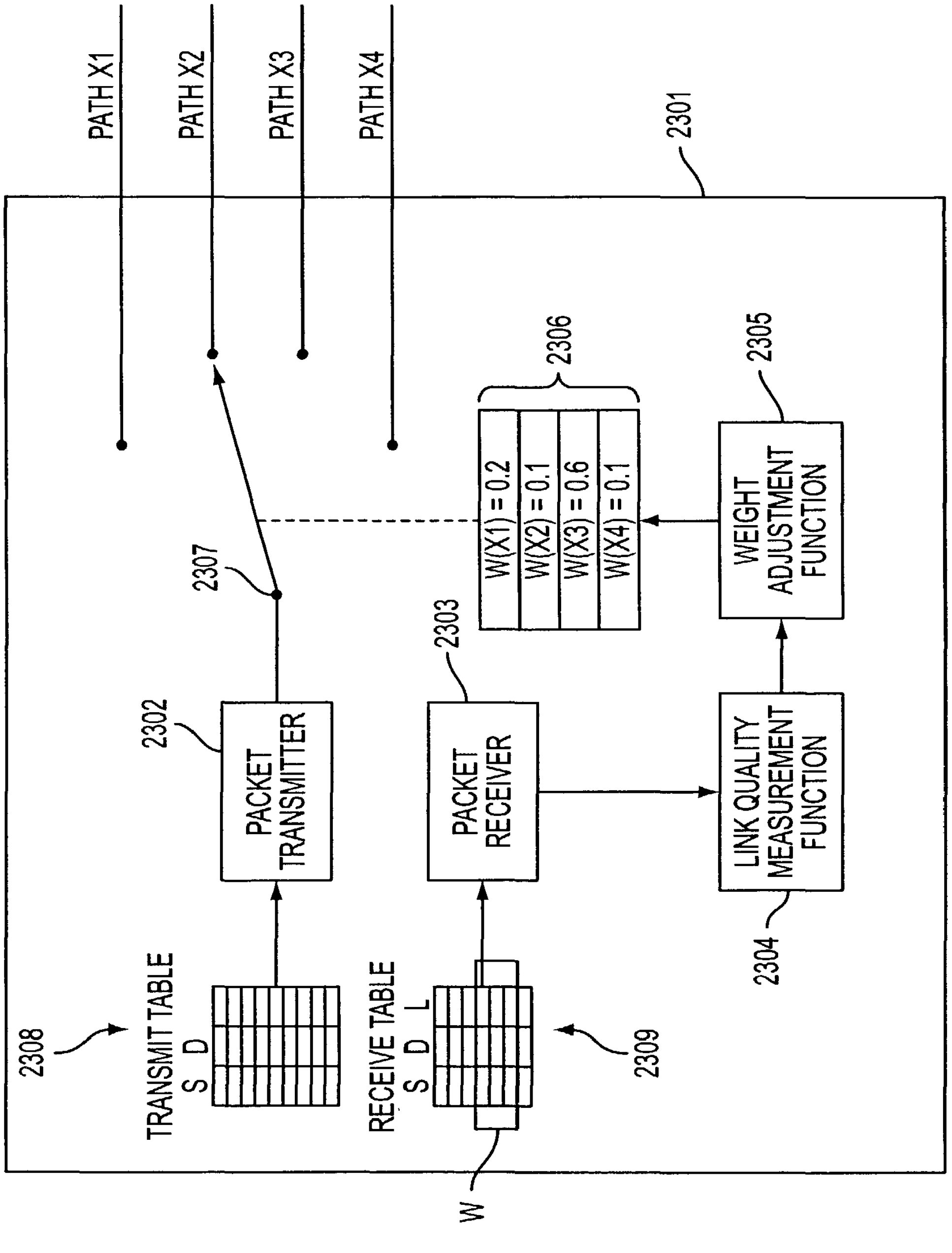

PATH X1
PATH X2
PATH X3
PATH X4
2301
2306
2305
2307
W(X1) = 0.2
W(X2) = 0.1
W(X3) = 0.6
W(X4) = 0.1
WEIGHT ADJUSTMENT FUNCTION
2303
2302
PACKET TRANSMITTER
PACKET RECEIVER
LINK QUALITY MEASUREMENT FUNCTION
2304
2308
TRANSMIT TABLE
S D
RECEIVE TABLE
S D L
2309
W

FIG. 23

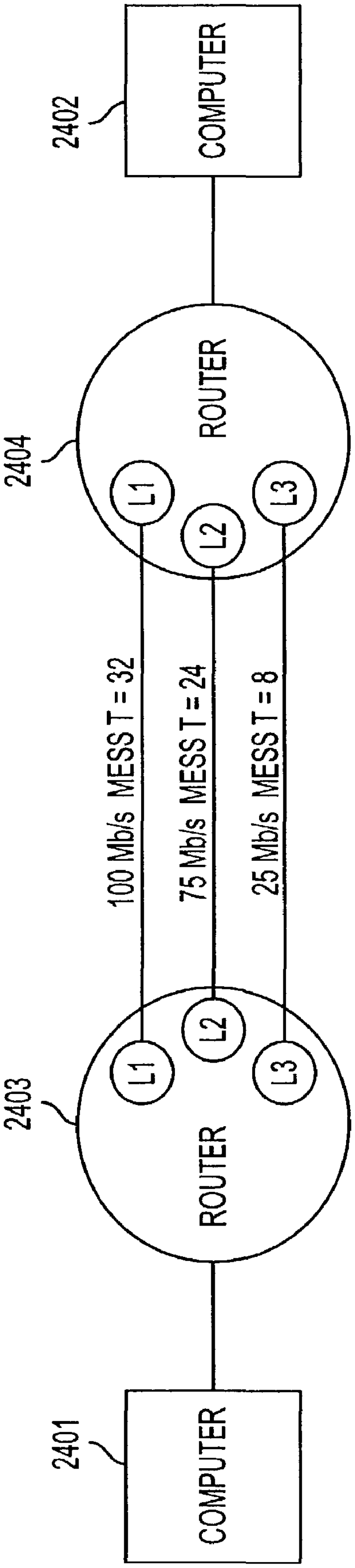

2402
COMPUTER
2404
ROUTER
L1
L2
L3
100 Mb/s MESS T = 32
75 Mb/s MESS T = 24
25 Mb/s MESS T = 8
L1
L2
L3
2403
ROUTER
2401
COMPUTER

FIG. 24

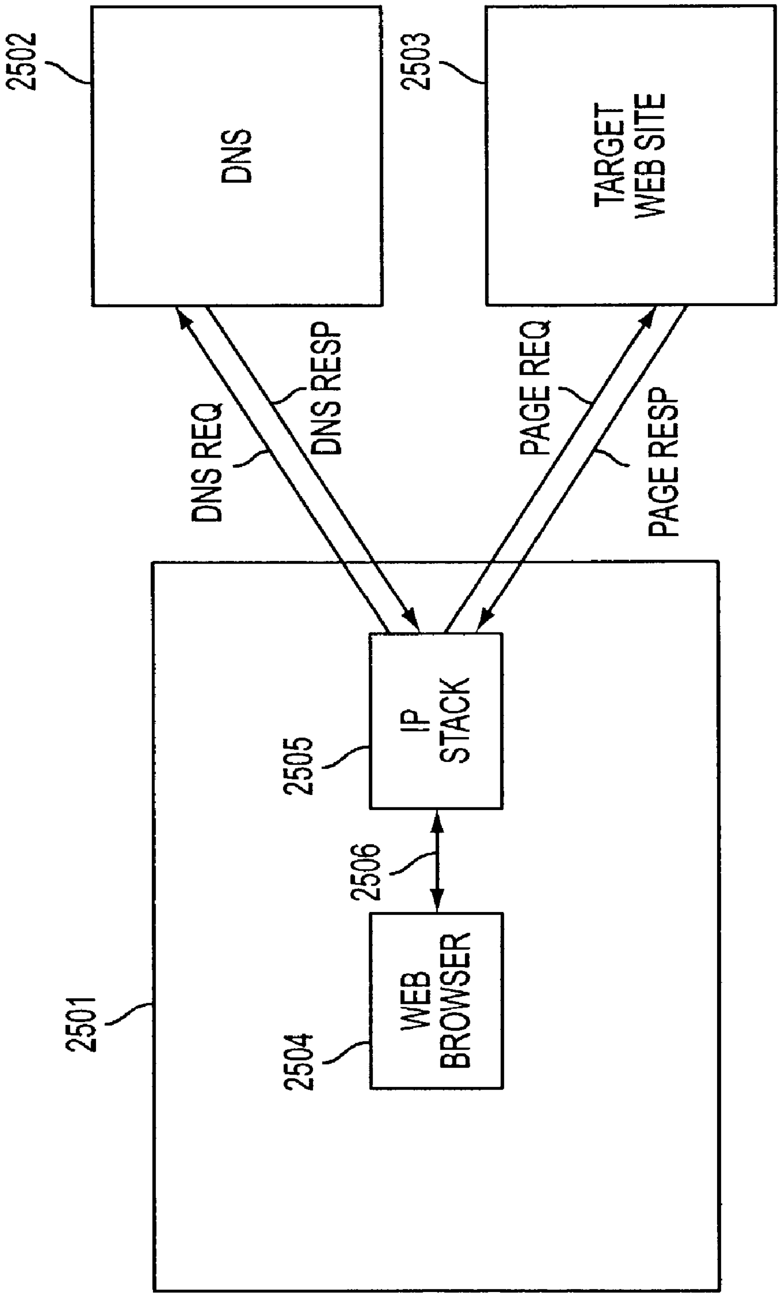


FIG. 25
(PRIOR ART)

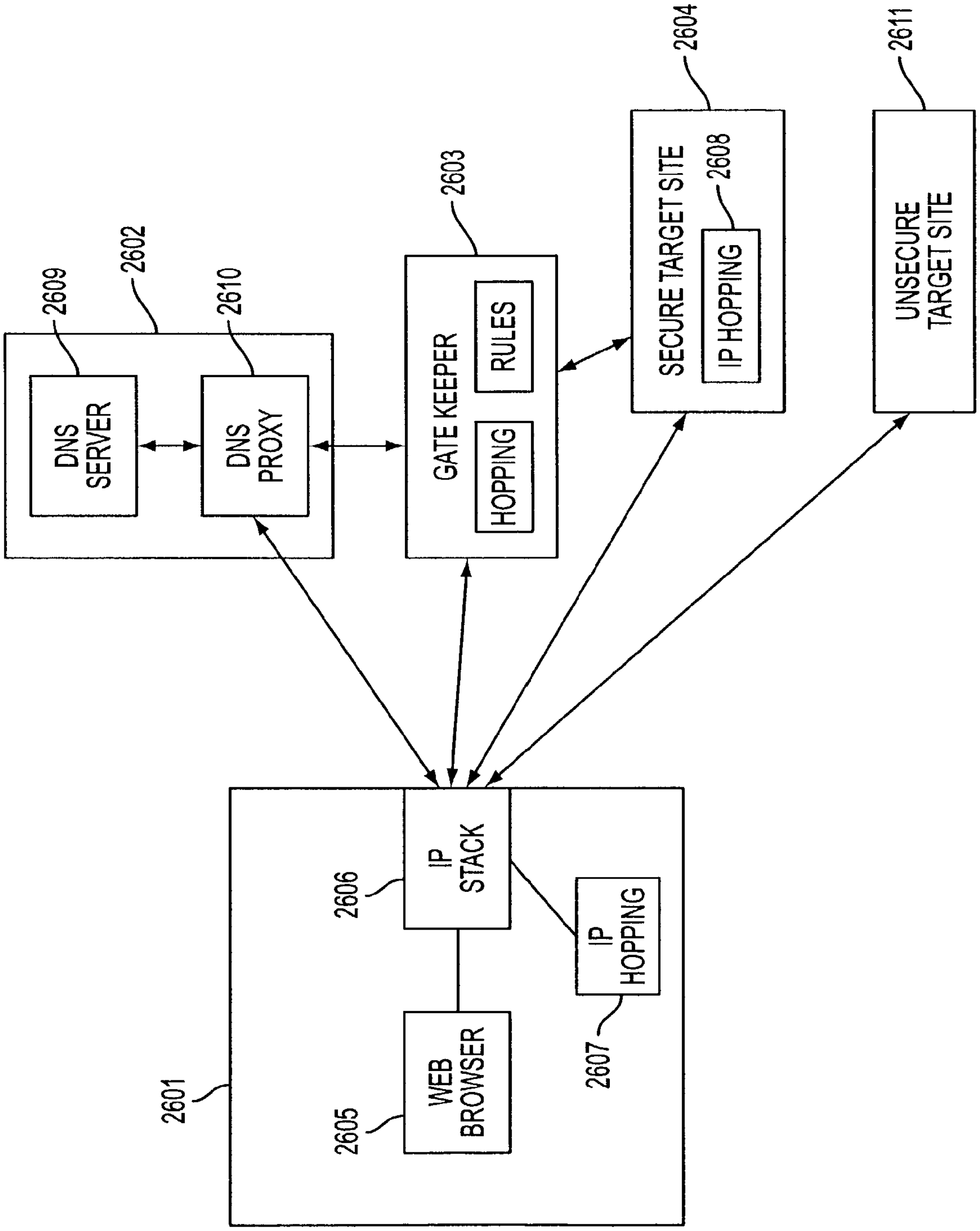

2609
2602
2610
DNS SERVER
DNS PROXY
2603
GATE KEEPER
RULES
HOPPING
2604
SECURE TARGET SITE
2608
IP HOPPING
2611
UNSECURE TARGET SITE
2606
IP STACK
IP HOPPING
2607
2601
WEB BROWSER
2605

FIG. 26

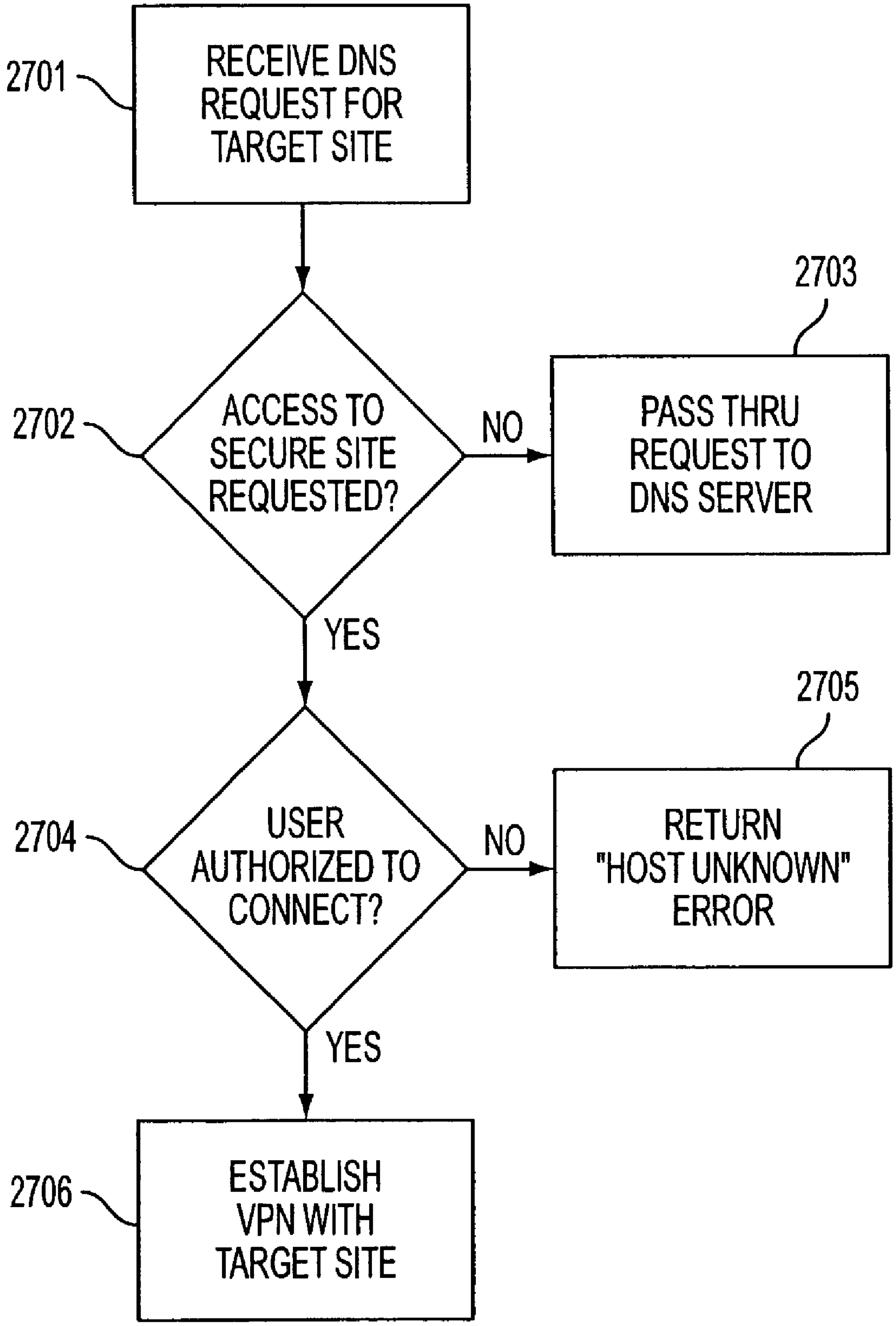

2701
RECEIVE DNS REQUEST FOR TARGET SITE
2702
ACCESS TO SECURE SITE REQUESTED?
NO
2703
PASS THRU REQUEST TO DNS SERVER
YES
2704
USER AUTHORIZED TO CONNECT?
NO
2705
RETURN "HOST UNKNOWN" ERROR
YES
2706
ESTABLISH VPN WITH TARGET SITE

FIG. 27

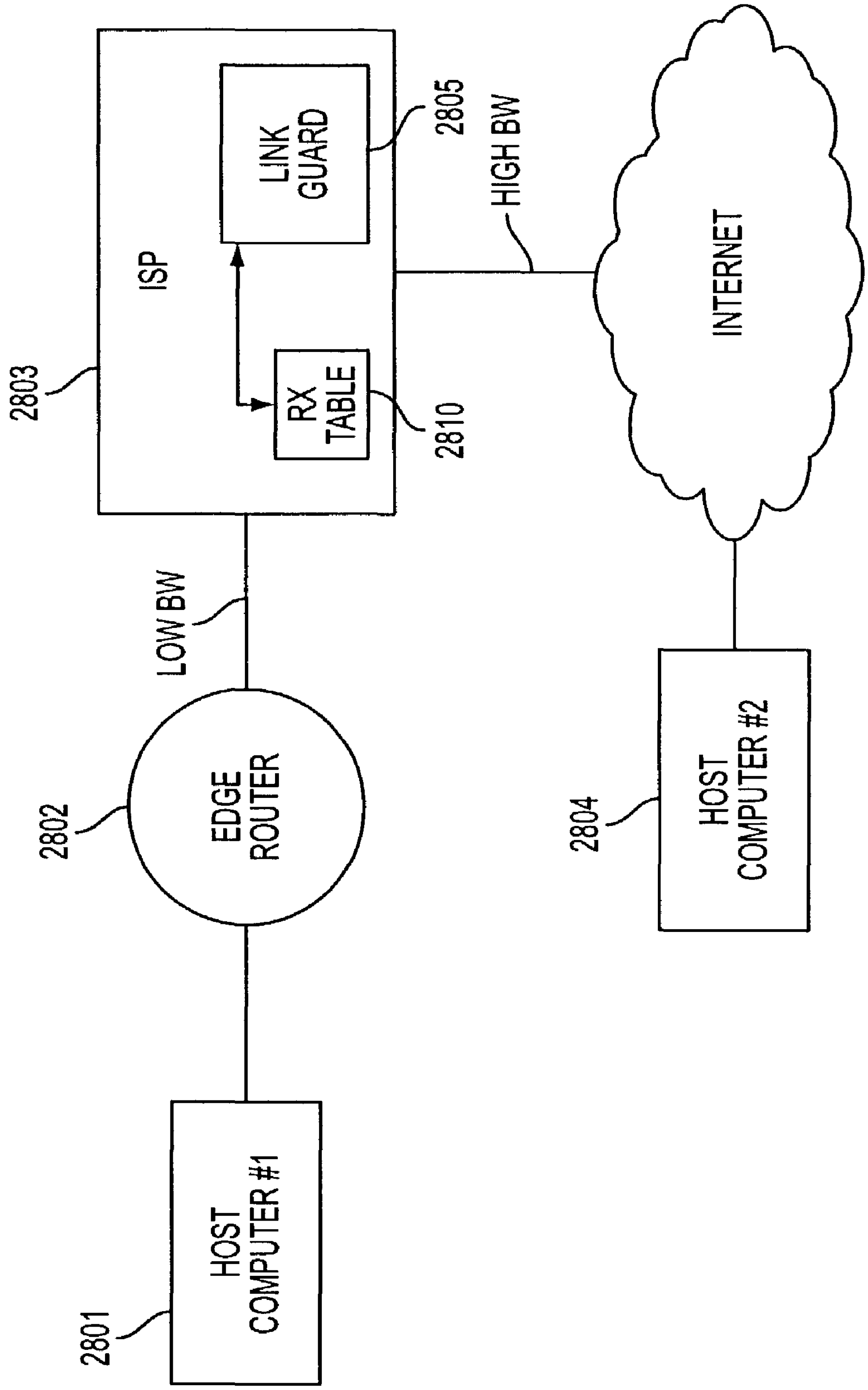

2803
ISP
LINK GUARD
2805
RX TABLE
2810
HIGH BW
INTERNET
LOW BW
2802
EDGE ROUTER
2804
HOST COMPUTER #2
2801
HOST COMPUTER #1

FIG. 28

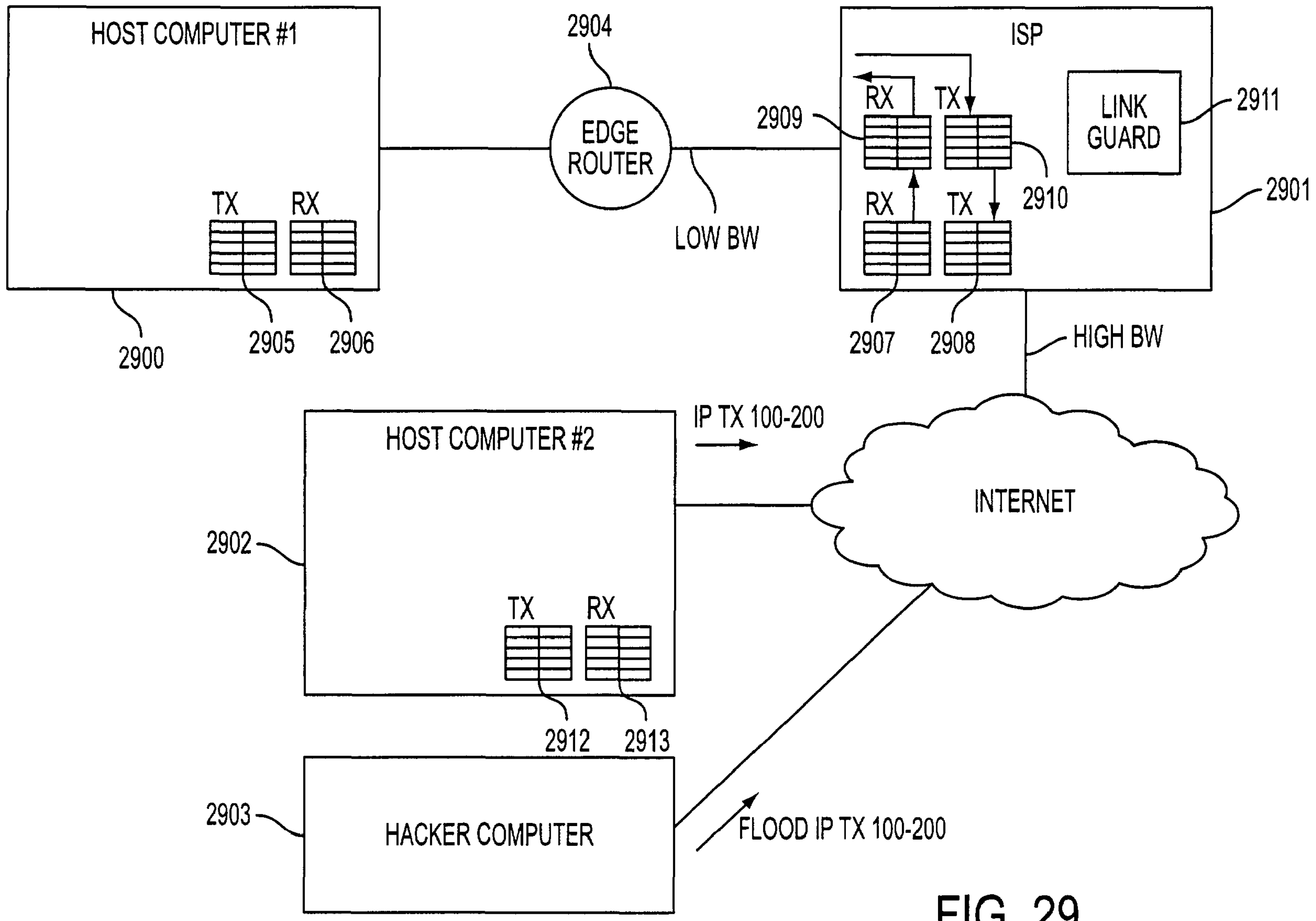

HOST COMPUTER #1
TX
RX
2900
2905
2906
2904
EDGE ROUTER
LOW BW
ISP
RX
TX
RX
TX
LINK GUARD
2909
2910
2911
2901
2907
2908
HIGH BW
HOST COMPUTER #2
IP TX 100-200
INTERNET
TX
RX
2902
2912
2913
2903
HACKER COMPUTER
FLOOD IP TX 100-200

FIG. 29

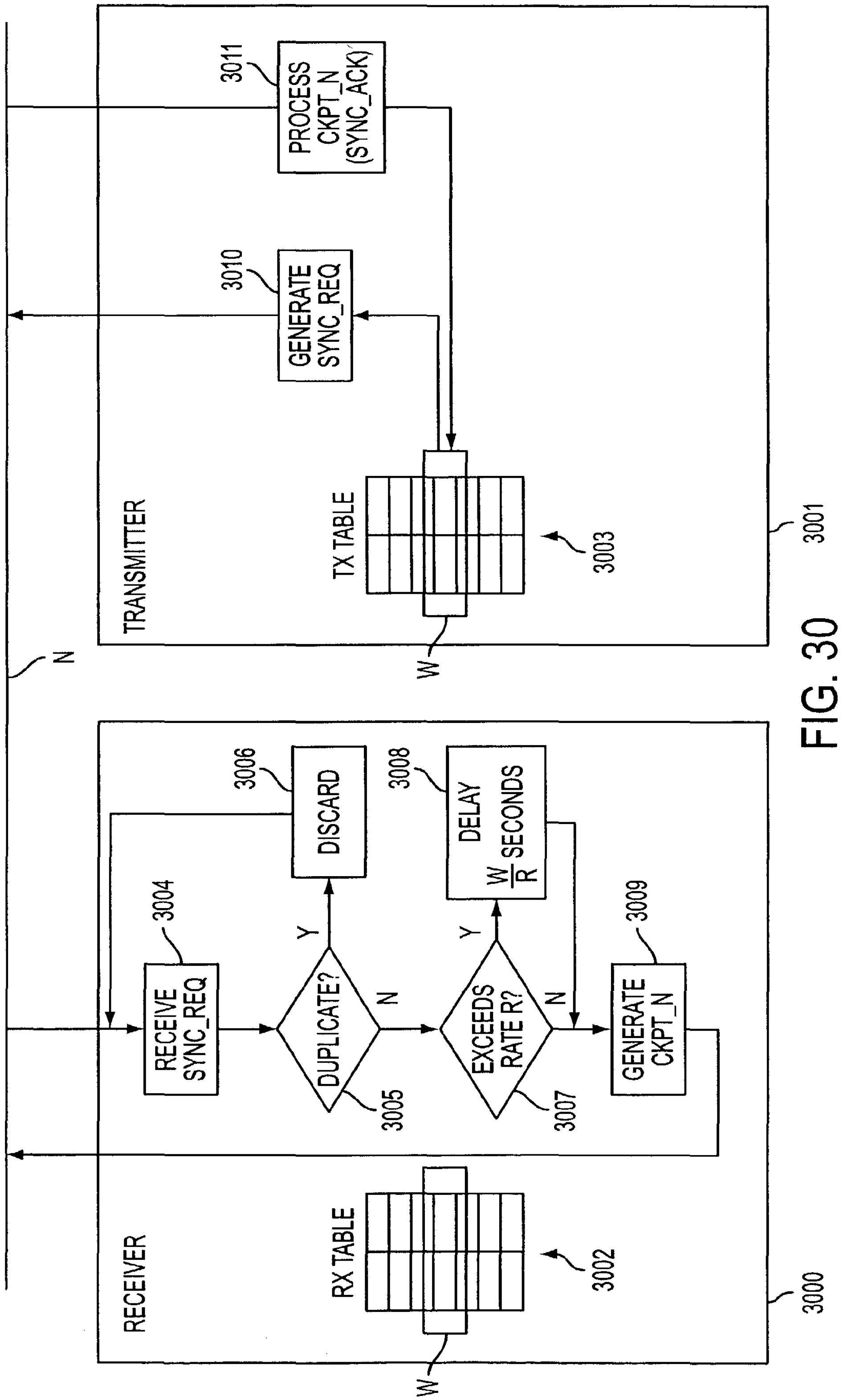

TRANSMITTER
3001
TX TABLE
3003
W
GENERATE SYNC_REQ
3010
PROCESS CKPT_N (SYNC_ACK)
3011
N
RECEIVER
3000
RX TABLE
3002
W
RECEIVE SYNC_REQ
3004
DUPLICATE?
3005
Y
N
DISCARD
3006
EXCEEDS RATE R?
3007
Y
N
DELAY W/R SECONDS
3008
GENERATE CKPT_N
3009

FIG. 30

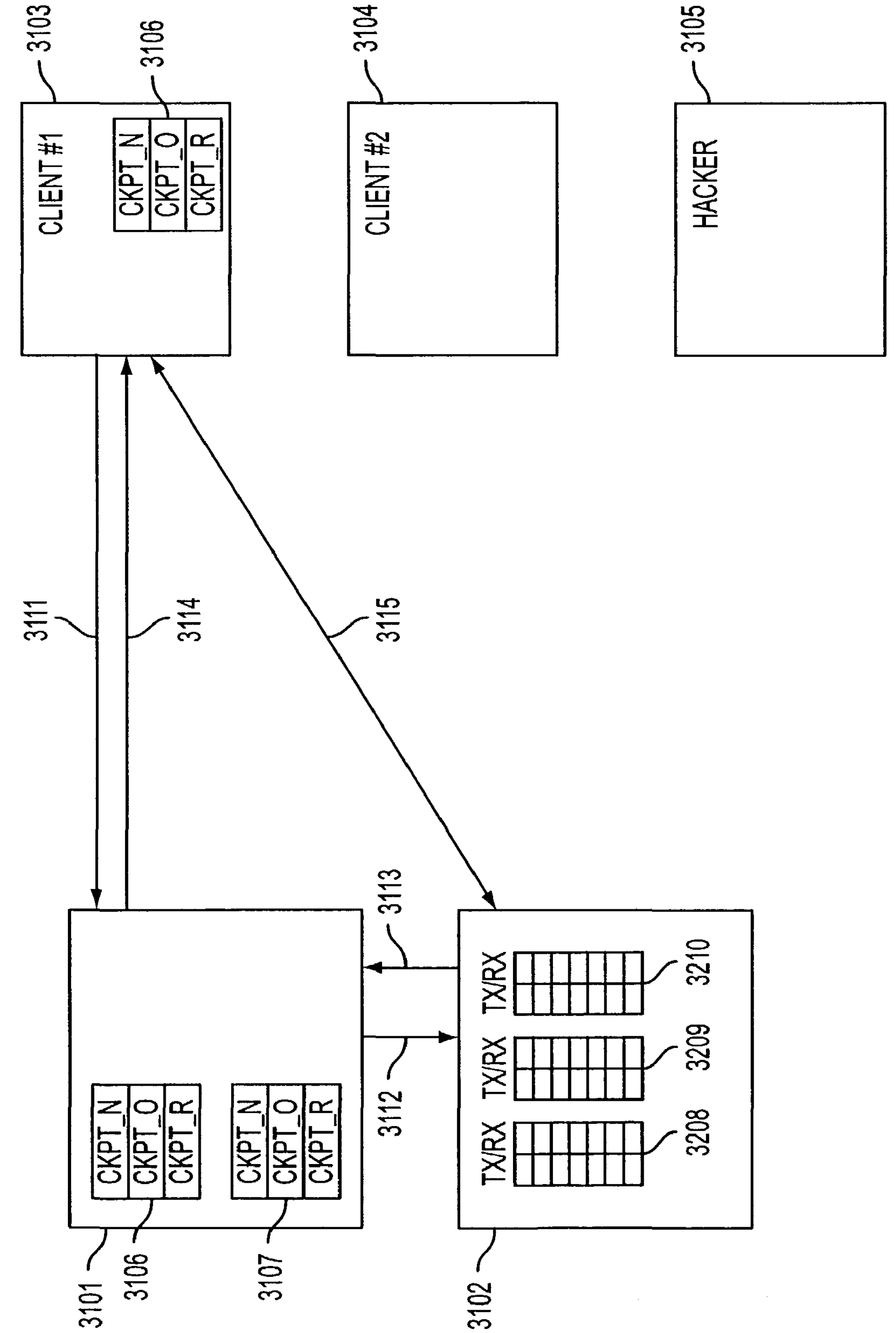

3103
3106
CLIENT #1
CKPT_N
CKPT_O
CKPT_R
3104
CLIENT #2
3105
HACKER
3111
3114
3115
3113
3112
CKPT_N
CKPT_O
CKPT_R
CKPT_N
CKPT_O
CKPT_R
TX/RX
TX/RX
TX/RX
3210
3209
3208
3101
3106
3107
3102

FIG. 31

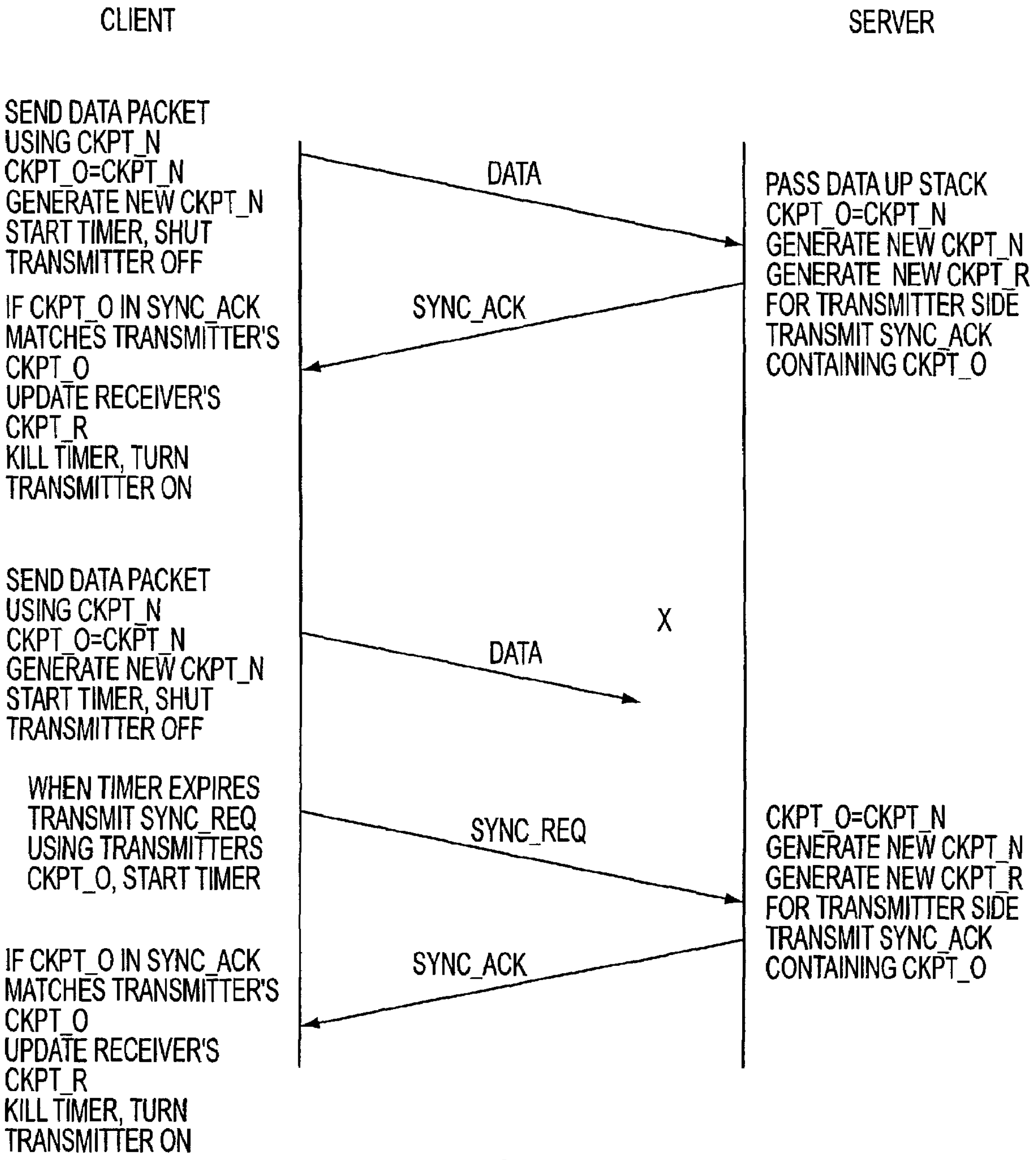

CLIENT
SERVER
SEND DATA PACKET USING CKPT_N CKPT_O=CKPT_N GENERATE NEW CKPT_N START TIMER, SHUT TRANSMITTER OFF
DATA
PASS DATA UP STACK CKPT_O=CKPT_N GENERATE NEW CKPT_N GENERATE NEW CKPT_R FOR TRANSMITTER SIDE TRANSMIT SYNC_ACK CONTAINING CKPT_O
SYNC_ACK
IF CKPT_O IN SYNC_ACK MATCHES TRANSMITTER'S CKPT_O UPDATE RECEIVER'S CKPT_R KILL TIMER, TURN TRANSMITTER ON
SEND DATA PACKET USING CKPT_N CKPT_O=CKPT_N GENERATE NEW CKPT_N START TIMER, SHUT TRANSMITTER OFF
X
DATA
WHEN TIMER EXPIRES TRANSMIT SYNC_REQ USING TRANSMITTERS CKPT_O, START TIMER
SYNC_REQ
CKPT_O=CKPT_N GENERATE NEW CKPT_N GENERATE NEW CKPT_R FOR TRANSMITTER SIDE TRANSMIT SYNC_ACK CONTAINING CKPT_O
SYNC_ACK
IF CKPT_O IN SYNC_ACK MATCHES TRANSMITTER'S CKPT_O UPDATE RECEIVER'S CKPT_R KILL TIMER, TURN TRANSMITTER ON

FIG. 32

ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a divisional application of 09/504,783 (filed Feb. 15, 2000), now U.S. Pat. No. 6,502,135, issued Dec. 31, 2002, which claims priority from and is a continuation-in-part of previously filed U.S. application Ser. No. 09/429,643 (filed Oct. 29, 1999) now U.S. Pat. No. 7,010,604. The subject matter of the '643 application, which is bodily incorporated herein, derives from provisional U.S. application No. 60/106,261 (filed Oct. 30, 1998) and 60/137,704 (filed Jun. 7, 1999).

GOVERNMENT CONTRACT RIGHTS

This invention was made with Government support under Contract No. 360000-1999-000000-QC-000-000 awarded by the Central Intelligence Agency. The Government has certain rights in the invention.

BACKGROUND OF THE INVENTION

A tremendous variety of methods have been proposed and implemented to provide security and anonymity for communications over the Internet. The variety stems, in part, from the different needs of different Internet users. A basic heuristic framework to aid in discussing these different security techniques is illustrated in FIG. **1**. Two terminals, an originating terminal **100** and a destination terminal **110** are in communication over the Internet. It is desired for the communications to be secure, that is, immune to eavesdropping. For example, terminal **100** may transmit secret information to terminal **110** over the Internet **107**. Also, it may be desired to prevent an eavesdropper from discovering that terminal **100** is in communication with terminal **110**. For example, if terminal **100** is a user and terminal **110** hosts a web site, terminal **100**'s user may not want anyone in the intervening networks to know what web sites he is "visiting." Anonymity would thus be an issue, for example, for companies that want to keep their market research interests private and thus would prefer to prevent outsiders from knowing which web-sites or other Internet resources they are "visiting." These two security issues may be called data security and anonymity, respectively.

Data security is usually tackled using some form of data encryption. An encryption key **48** is known at both the originating and terminating terminals **100** and **110**. The keys may be private and public at the originating and destination terminals **100** and **110**, respectively or they may be symmetrical keys (the same key is used by both parties to encrypt and decrypt). Many encryption methods are known and usable in this context.

To hide traffic from a local administrator or ISP, a user can employ a local proxy server in communicating over an encrypted channel with an outside proxy such that the local administrator or ISP only sees the encrypted traffic. Proxy servers prevent destination servers from determining the identities of the originating clients. This system employs an intermediate server interposed between client and destination server. The destination server sees only the Internet Protocol (IP) address of the proxy server and not the originating client. The target server only sees the address of the outside proxy. This scheme relies on a trusted outside proxy server. Also, proxy schemes are vulnerable to traffic analysis methods of determining identities of transmitters and receivers. Another important limitation of proxy servers is that the server knows the identities of both calling and called parties. In many instances, an originating terminal, such as terminal A, would prefer to keep its identity concealed from the proxy, for example, if the proxy server is provided by an Internet service provider (ISP).

To defeat traffic analysis, a scheme called Chaum's mixes employs a proxy server that transmits and receives fixed length messages, including dummy messages. Multiple originating terminals are connected through a mix (a server) to multiple target servers. It is difficult to tell which of the originating terminals are communicating to which of the connected target servers, and the dummy messages confuse eavesdroppers' efforts to detect communicating pairs by analyzing traffic. A drawback is that there is a risk that the mix server could be compromised. One way to deal with this risk is to spread the trust among multiple mixes. If one mix is compromised, the identities of the originating and target terminals may remain concealed. This strategy requires a number of alternative mixes so that the intermediate servers interposed between the originating and target terminals are not determinable except by compromising more than one mix. The strategy wraps the message with multiple layers of encrypted addresses. The first mix in a sequence can decrypt only the outer layer of the message to reveal the next destination mix in sequence. The second mix can decrypt the message to reveal the next mix and so on. The target server receives the message and, optionally, a multi-layer encrypted payload containing return information to send data back in the same fashion. The only way to defeat such a mix scheme is to collude among mixes. If the packets are all fixed-length and intermixed with dummy packets, there is no way to do any kind of traffic analysis.

Still another anonymity technique, called 'crowds,' protects the identity of the originating terminal from the intermediate proxies by providing that originating terminals belong to groups of proxies called crowds. The crowd proxies are interposed between originating and target terminals. Each proxy through which the message is sent is randomly chosen by an upstream proxy. Each intermediate proxy can send the message either to another randomly chosen proxy in the "crowd" or to the destination. Thus, even crowd members cannot determine if a preceding proxy is the originator of the message or if it was simply passed from another proxy.

ZKS (Zero-Knowledge Systems) Anonymous IP Protocol allows users to select up to any of five different pseudonyms, while desktop software encrypts outgoing traffic and wraps it in User Datagram Protocol (UDP) packets. The first server in a 2+-hop system gets the UDP packets, strips off one layer of encryption to add another, then sends the traffic to the next server, which strips off yet another layer of encryption and adds a new one. The user is permitted to control the number of hops. At the final server, traffic is decrypted with an untraceable IP address. The technique is called onion-routing. This method can be defeated using traffic analysis. For a simple example, bursts of packets from a user during low-duty periods can reveal the identities of sender and receiver.

Firewalls attempt to protect LANs from unauthorized access and hostile exploitation or damage to computers connected to the LAN. Firewalls provide a server through which all access to the LAN must pass. Firewalls are centralized systems that require administrative overhead to maintain. They can be compromised by virtual-machine applications ("applets"). They instill a false sense of security that leads to security breaches for example by users sending sensitive information to servers outside the firewall or encouraging use of modems to sidestep the firewall security. Firewalls are not useful for distributed systems such as business travelers, extranets, small teams, etc.

SUMMARY OF THE INVENTION

A secure mechanism for communicating over the internet, including a protocol referred to as the Tunneled Agile Routing Protocol (TARP), uses a unique two-layer encryption format and special TARP routers. TARP routers are similar in function to regular IP routers. Each TARP router has one or more IP addresses and uses normal IP protocol to send IP packet messages ("packets" or "datagrams"). The IP packets exchanged between TARP terminals via TARP routers are actually encrypted packets whose true destination address is concealed except to TARP routers and servers. The normal or "clear" or "outside" IP header attached to TARP IP packets contains only the address of a next hop router or destination server. That is, instead of indicating a final destination in the destination field of the IP header, the TARP packet's IP header always points to a next-hop in a series of TARP router hops, or to the final destination. This means there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet since the destination could always be next-hop TARP router as well as the final destination.

Each TARP packet's true destination is concealed behind a layer of encryption generated using a link key. The link key is the encryption key used for encrypted communication between the hops intervening between an originating TARP terminal and a destination TARP terminal. Each TARP router can remove the outer layer of encryption to reveal the destination router for each TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal by the sender/receiver IP numbers in the cleartext IP header.

Once the outer layer of encryption is removed, the TARP router determines the final destination. Each TARP packet **140** undergoes a minimum number of hops to help foil traffic analysis. The hops may be chosen at random or by a fixed value. As a result, each TARP packet may make random trips among a number of geographically disparate routers before reaching its destination. Each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined. This feature is called agile routing. The fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. The associated advantages have to do with the inner layer of encryption discussed below. Agile routing is combined with another feature that furthers this purpose; a feature that ensures that any message is broken into multiple packets.

The IP address of a TARP router can be changed, a feature called IP agility. Each TARP router, independently or under direction from another TARP terminal or router, can change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs.

The message payload is hidden behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the intervening TARP routers. The session key is used to decrypt the payloads of the TARP packets permitting the data stream to be reconstructed.

Communication may be made private using link and session keys, which in turn may be shared and used according to any desired method. For example, public/private keys or symmetric keys may be used.

To transmit a data stream, a TARP originating terminal constructs a series of TARP packets from a series of IP packets generated by a network (IP) layer process. (Note that the terms "network layer," "data link layer," "application layer," etc. used in this specification correspond to the Open Systems Interconnection (OSI) network terminology.) The payloads of these packets are assembled into a block and chain-block encrypted using the session key. This assumes, of course, that all the IP packets are destined for the same TARP terminal. The block is then interleaved and the interleaved encrypted block is broken into a series of payloads, one for each TARP packet to be generated. Special TARP headers IPT are then added to each payload using the IP headers from the data stream packets. The TARP headers can be identical to normal IP headers or customized in some way. They should contain a formula or data for deinterleaving the data at the destination TARP terminal, a time-to-live (TTL) parameter to indicate the number of hops still to be executed, a data type identifier which indicates whether the payload contains, for example, TCP or UDP data, the sender's TARP address, the destination TARP address, and an indicator as to whether the packet contains real or decoy data or a formula for filtering out decoy data if decoy data is spread in some way through the TARP payload data.

Note that although chain-block encryption is discussed here with reference to the session key, any encryption method may be used. Preferably, as in chain block encryption, a method should be used that makes unauthorized decryption difficult without an entire result of the encryption process. Thus, by separating the encrypted block among multiple packets and making it difficult for an interloper to obtain access to all of such packets, the contents of the communications are provided an extra layer of security.

Decoy or dummy data can be added to a stream to help foil traffic analysis by reducing the peak-to-average network load. It may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to portion, or entirety, of a message, and that portion or entirety then interleaved into a number of separate packets. Considering the agile IP routing of the packets, and the attendant difficulty of reconstructing an entire sequence of packets to form a single block-encrypted message element, decoy packets can significantly increase the difficulty of reconstructing an entire data stream.

The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Because the encryption system described above is insertable between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the Network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicating that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. In addition, it may create a subprocess that maintains the original IP address and continues interacting with the attacker in some manner.

Decoy packets may be generated by each TARP terminal on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis.

In various other embodiments of the invention, a scalable version of the system may be constructed in which a plurality of IP addresses are preassigned to each pair of communicating nodes in the network. Each pair of nodes agrees upon an algorithm for "hopping" between IP addresses (both sending and receiving), such that an eavesdropper sees apparently continuously random IP address pairs (source and destination) for packets transmitted between the pair. Overlapping or "reusable" IP addresses may be allocated to different users on the same subnet, since each node merely verifies that a particular packet includes a valid source/destination pair from the agreed-upon algorithm. Source/destination pairs are preferably not reused between any two nodes during any given end-to-end session, though limited IP block sizes or lengthy sessions might require it.

Further improvements described in this continuation-in-part application include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is an illustration of secure communications over the Internet according to a prior art embodiment.

FIG. **2** is an illustration of secure communications over the Internet according to a an embodiment of the invention.

FIG. **3***a* is an illustration of a process of forming a tunneled IP packet according to an embodiment of the invention.

FIG. **3***b* is an illustration of a process of forming a tunneled IP packet according to another embodiment of the invention.

FIG. **4** is an illustration of an OSI layer location of processes that may be used to implement the invention.

FIG. **5** is a flow chart illustrating a process for routing a tunneled packet according to an embodiment of the invention.

FIG. **6** is a flow chart illustrating a process for forming a tunneled packet according to an embodiment of the invention.

FIG. **7** is a flow chart illustrating a process for receiving a tunneled packet according to an embodiment of the invention.

FIG. **8** shows how a secure session is established and synchronized between a client and a TARP router.

FIG. **9** shows an IP address hopping scheme between a client computer and TARP router using transmit and receive tables in each computer.

FIG. **10** shows physical link redundancy among three Internet Service Providers (ISPs) and a client computer.

FIG. **11** shows how multiple IP packets can be embedded into a single "frame" such as an Ethernet frame, and further shows the use of a discriminator field to camouflage true packet recipients.

FIG. **12**A shows a system that employs hopped hardware addresses, hopped IP addresses, and hopped discriminator fields.

FIG. **12**B shows several different approaches for hopping hardware addresses, IP addresses, and discriminator fields in combination.

FIG. **13** shows a technique for automatically re-establishing synchronization between sender and receiver through the use of a partially public sync value.

FIG. **14** shows a "checkpoint" scheme for regaining synchronization between a sender and recipient.

FIG. **15** shows further details of the checkpoint scheme of FIG. **14**.

FIG. **16** shows how two addresses can be decomposed into a plurality of segments for comparison with presence vectors.

FIG. **17** shows a storage array for a receiver's active addresses.

FIG. **18** shows the receiver's storage array after receiving a sync request.

FIG. **19** shows the receiver's storage array after new addresses have been generated.

FIG. **20** shows a system employing distributed transmission paths.

FIG. **21** shows a plurality of link transmission tables that can be used to route packets in the system of FIG. **20**.

FIG. **22**A shows a flowchart for adjusting weight value distributions associated with a plurality of transmission links.

FIG. **22**B shows a flowchart for setting a weight value to zero if a transmitter turns off.

FIG. **23** shows a system employing distributed transmission paths with adjusted weight value distributions for each path.

FIG. **24** shows an example using the system of FIG. **23**.

FIG. **25** shows a conventional domain-name look-up service.

FIG. **26** shows a system employing a DNS proxy server with transparent VPN creation.

FIG. **27** shows steps that can be carried out to implement transparent VPN creation based on a DNS look-up function.

FIG. **28** shows a system including a link guard function that prevents packet overloading on a low-bandwidth link LOW BW.

FIG. **29** shows one embodiment of a system employing the principles of FIG. **28**.

FIG. **30** shows a system that regulates packet transmission rates by throttling the rate at which synchronizations are performed.

FIG. **31** shows a signaling server **3101** and a transport server **3102** used to establish a VPN with a client computer.

FIG. **32** shows message flows relating to synchronization protocols of FIG. **31**.

DETAILED DESCRIPTION OF THE INVENTION

Referring to FIG. **2**, a secure mechanism for communicating over the internet employs a number of special routers or servers, called TARP routers **122-127** that are similar to regular IP routers **128-132** in that each has one or more IP addresses and uses normal IP protocol to send normal-looking IP packet messages, called TARP packets **140**. TARP packets **140** are identical to normal IP packet messages that are routed by regular IP routers **128-132** because each TARP packet **140** contains a destination address as in a normal IP packet. However, instead of indicating a final destination in the destination field of the IP header, the TARP packet's **140** IP header always points to a next-hop in a series of TARP router hops, or the final destination, TARP terminal **110**. Because the header of the TARP packet contains only the next-hop destination, there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet **140** since the destination could always be the next-hop TARP router as well as the final destination, TARP terminal **110**.

Each TARP packet's true destination is concealed behind an outer layer of encryption generated using a link key **146**. The link key **146** is the encryption key used for encrypted communication between the end points (TARP terminals or TARP routers) of a single link in the chain of hops connecting the originating TARP terminal **100** and the destination TARP terminal **110**. Each TARP router **122-127**, using the link key **146** it uses to communicate with the previous hop in a chain, can use the link key to reveal the true destination of a TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal (which may indicate the link key used) by the sender field of the clear IP header. Alternatively, this identity may be hidden behind another layer of encryption in available bits in the clear IP header. Each TARP router, upon receiving a TARP message, determines if the message is a TARP message by using authentication data in the TARP packet. This could be recorded in available bytes in the TARP packet's IP header. Alternatively, TARP packets could be authenticated by attempting to decrypt using the link key **146** and determining if the results are as expected. The former may have computational advantages because it does not involve a decryption process.

Once the outer layer of decryption is completed by a TARP router **122-127**, the TARP router determines the final destination. The system is preferably designed to cause each TARP packet **140** to undergo a minimum number of hops to help foil traffic analysis. The time to live counter in the IP header of the TARP message may be used to indicate a number of TARP router hops yet to be completed. Each TARP router then would decrement the counter and determine from that whether it should forward the TARP packet **140** to another TARP router **122-127** or to the destination TARP terminal **110**. If the time to live counter is zero or below zero after decrementing, for an example of usage, the TARP router receiving the TARP packet **140** may forward the TARP packet **140** to the destination TARP terminal **110**. If the time to live counter is above zero after decrementing, for an example of usage, the TARP router receiving the TARP packet **140** may forward the TARP packet **140** to a TARP router **122-127** that the current TARP terminal chooses at random. As a result, each TARP packet **140** is routed through some minimum number of hops of TARP routers **122-127** which are chosen at random.

Thus, each TARP packet, irrespective of the traditional factors determining traffic in the Internet, makes random trips among a number of geographically disparate routers before reaching its destination and each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined as described above. This feature is called agile routing. For reasons that will become clear shortly, the fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. Agile routing is combined with another feature that furthers this purpose, a feature that ensures that any message is broken into multiple packets.

A TARP router receives a TARP packet when an IP address used by the TARP router coincides with the IP address in the TARP packet's IP header IP_C. The IP address of a TARP router, however, may not remain constant. To avoid and manage attacks, each TARP router, independently or under direction from another TARP terminal or router, may change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs. In reality, whenever a TARP router looks up the address of a destination in the encrypted header, it must convert a TARP address to a real IP address using its LUT.

While every TARP router receiving a TARP packet has the ability to determine the packet's final destination, the message payload is embedded behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the TARP routers **122-127** intervening between the originating **100** and destination **110** TARP terminals. The session key is used to decrypt the payloads of the TARP packets **140** permitting an entire message to be reconstructed.

In one embodiment, communication may be made private using link and session keys, which in turn may be shared and used according any desired method. For example, a public key or symmetric keys may be communicated between link or session endpoints using a public key method. Any of a variety of other mechanisms for securing data to ensure that only authorized computers can have access to the private information in the TARP packets **140** may be used as desired.

Referring to FIG. **3***a*, to construct a series of TARP packets, a data stream **300** of IP packets **207***a*, **207***b*, **207***c*, etc., such series of packets being formed by a network (IP) layer process, is broken into a series of small sized segments. In the present example, equal-sized segments **1-9** are defined and used to construct a set of interleaved data packets A, B, and C. Here it is assumed that the number of interleaved packets A, B, and C formed is three and that the number of IP packets **207***a*-**207***c* used to form the three interleaved packets A, B, and C is exactly three. Of course, the number of IP packets spread over a group of interleaved packets may be any convenient number as may be the number of interleaved packets over which the incoming data stream is spread. The latter, the number of interleaved packets over which the data stream is spread, is called the interleave window.

To create a packet, the transmitting software interleaves the normal IP packets **207***a* et. seq. to form a new set of interleaved payload data **320**. This payload data **320** is then encrypted using a session key to form a set of session-key-encrypted payload data **330**, each of which, A, B, and C, will form the payload of a TARP packet. Using the IP header data, from the original packets **207***a*-**207***c*, new TARP headers IPT are formed. The TARP headers IPT can be identical to normal IP headers or customized in some way. In a preferred embodiment, the TARP headers IPT are IP headers with added data providing the following information required for routing and reconstruction of messages, some of which data is ordinarily, or capable of being, contained in normal IP headers:

1. A window sequence number—an identifier that indicates where the packet belongs in the original message sequence.

2. An interleave sequence number—an identifier that indicates the interleaving sequence used to form the packet so that the packet can be deinterleaved along with other packets in the interleave window.

3. A time-to-live (TTL) datum—indicates the number of TARP-router-hops to be executed before the packet reaches its destination. Note that the TTL parameter may provide a datum to be used in a probabilistic formula for determining whether to route the packet to the destination or to another hop.

4. Data type identifier—indicates whether the payload contains, for example, TCP or UDP data.

5. Sender's address—indicates the sender's address in the TARP network.

6. Destination address—indicates the destination terminal's address in the TARP network.

7. Decoy/Real—an indicator of whether the packet contains real message data or dummy decoy data or a combination.

Obviously, the packets going into a single interleave window must include only packets with a common destination. Thus, it is assumed in the depicted example that the IP headers of IP packets **207***a*-**207***c* all contain the same destination address or at least will be received by the same terminal so that they can be deinterleaved. Note that dummy or decoy data or packets can be added to form a larger interleave window than would otherwise be required by the size of a given message. Decoy or dummy data can be added to a stream to help foil traffic analysis by leveling the load on the network. Thus, it may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to a portion, or the entirety, of a message, and that portion or entirety then interleaved into a number of separate packets.

Referring to FIG. **3***b*, in an alternative mode of TARP packet construction, a series of IP packets are accumulated to make up a predefined interleave window. The payloads of the packets are used to construct a single block **520** for chain block encryption using the session key. The payloads used to form the block are presumed to be destined for the same terminal. The block size may coincide with the interleave window as depicted in the example embodiment of FIG. **3***b*. After encryption, the encrypted block is broken into separate payloads and segments which are interleaved as in the embodiment of FIG. **3***a*. The resulting interleaved packets A, B, and C, are then packaged as TARP packets with TARP headers as in the Example of FIG. **3***a*. The remaining process is as shown in, and discussed with reference to, FIG. **3***a*.

Once the TARP packets **340** are formed, each entire TARP packet **340**, including the TARP header IP_T, is encrypted using the link key for communication with the first-hop-TARP router. The first hop TARP router is randomly chosen. A final unencrypted IP header IP_C is added to each encrypted TARP packet **340** to form a normal IP packet **360** that can be transmitted to a TARP router. Note that the process of constructing the TARP packet **360** does not have to be done in stages as described. The above description is just a useful heuristic for describing the final product, namely, the TARP packet.

Note that, TARP header IP_T could be a completely custom header configuration with no similarity to a normal IP header except that it contain the information identified above. This is so since this header is interpreted by only TARP routers.

The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Referring to FIG. **4**, a TARP transceiver **405** can be an originating terminal **100**, a destination terminal **110**, or a TARP router **122-127**. In each TARP Transceiver **405**, a transmitting process is generated to receive normal packets from the Network (IP) layer and generate TARP packets for communication over the network. A receiving process is generated to receive normal IP packets containing TARP packets and generate from these normal IP packets which are "passed up" to the Network (IP) layer. Note that where the TARP Transceiver **405** is a router, the received TARP packets **140** are not processed into a stream of IP packets **415** because they need only be authenticated as proper TARP packets and then passed to another TARP router or a TARP destination terminal **110**. The intervening process, a "TARP Layer" **420**, could be combined with either the data link layer **430** or the Network layer **410**. In either case, it would intervene between the data link layer **430** so that the process would receive regular IP packets containing embedded TARP packets and "hand up" a series of reassembled IP packets to the Network layer **410**. As an example of combining the TARP layer **420** with the data link layer **430**, a program may augment the normal processes running a communications card, for example, an Ethernet card. Alternatively, the TARP layer processes may form part of a dynamically loadable module that is loaded and executed to support communications between the network and data link layers.

Because the encryption system described above can be inserted between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

Note that IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicates that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. To accomplish this, the TARP process will construct a TARP-formatted message, in the style of Internet Control Message Protocol (ICMP) datagrams as an example; this message will contain the machine's TARP address, its previous IP address, and its new IP address. The TARP layer will transmit this packet to at least one known TARP router; then upon receipt and validation of the message, the TARP router will update its LUT with the new IP address for the stated TARP address. The TARP router will then format a similar message, and broadcast it to the other TARP routers so that they may update their LUTs. Since the total number of TARP routers on any given subnet is expected to be relatively small, this process of updating the LUTs should be relatively fast. It may not, however, work as well when there is a relatively large number of TARP routers and/or a relatively large number of clients; this has motivated a refinement of this architecture to provide scalability; this refinement has led to a second embodiment, which is discussed below.

Upon detection of an attack, the TARP process may also create a subprocess that maintains the original IP address and continues interacting with the attacker. The latter may provide an opportunity to trace the attacker or study the attacker's methods (called "fishbowling" drawing upon the analogy of a small fish in a fish bowl that "thinks" it is in the ocean but is actually under captive observation). A history of the communication between the attacker and the abandoned (fishbowled) IP address can be recorded or transmitted for human analysis or further synthesized for purposes of responding in some way.

As mentioned above, decoy or dummy data or packets can be added to outgoing data streams by TARP terminals or routers. In addition to making it convenient to spread data over a larger number of separate packets, such decoy packets can also help to level the load on inactive portions of the Internet to help foil traffic analysis efforts.

Decoy packets may be generated by each TARP terminal **100**, **110** or each router **122**-**127** on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. That is, when a series of messages are received, the decoy packet generation rate may be increased. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis. The rate of reception of packets, decoy or otherwise, may be indicated to the decoy packet dropping and generating processes through perishable decoy and regular packet counters. (A perishable counter is one that resets or decrements its value in response to time so that it contains a high value when it is incremented in rapid succession and a small value when incremented either slowly or a small number of times in rapid succession.) Note that destination TARP terminal **110** may generate decoy packets equal in number and size to those TARP packets received to make it appear it is merely routing packets and is therefore not the destination terminal.

Referring to FIG. **5**, the following particular steps may be employed in the above-described method for routing TARP packets.

S0. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.

S2. The TARP packet may be probed in some way to authenticate the packet before attempting to decrypt it using the link key. That is, the router may determine that the packet is an authentic TARP packet by performing a selected operation on some data included with the clear IP header attached to the encrypted TARP packet contained in the payload. This makes it possible to avoid performing decryption on packets that are not authentic TARP packets.

S3. The TARP packet is decrypted to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.

S4. If the packet is a decoy packet, the perishable decoy counter is incremented.

S5. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the router may choose to throw it away. If the received packet is a decoy packet and it is determined that it should be thrown away (S6), control returns to step S0.

S7. The TTL parameter of the TARP header is decremented and it is determined if the TTL parameter is greater than zero.

S8. If the TTL parameter is greater than zero, a TARP address is randomly chosen from a list of TARP addresses maintained by the router and the link key and IP address corresponding to that TARP address memorized for use in creating a new IP packet containing the TARP packet.

S9. If the TTL parameter is zero or less, the link key and IP address corresponding to the TARP address of the destination are memorized for use in creating the new IP packet containing the TARP packet.

S10. The TARP packet is encrypted using the memorized link key.

S11. An IP header is added to the packet that contains the stored IP address, the encrypted TARP packet wrapped with an IP header, and the completed packet transmitted to the next hop or destination.

Referring to FIG. **6**, the following particular steps may be employed in the above-described method for generating TARP packets.

S20. A background loop operation applies an algorithm that determines the generation of decoy IP packets. The loop is interrupted when a data stream containing IP packets is received for transmission.

S21. The received IP packets are grouped into a set consisting of messages with a constant IP destination address. The set is further broken down to coincide with a maximum size of an interleave window The set is encrypted, and interleaved into a set of payloads destined to become TARP packets.

S22. The TARP address corresponding to the IP address is determined from a lookup table and stored to generate the TARP header. An initial TTL count is generated and stored in the header. The TTL count may be random with minimum and maximum values or it may be fixed or determined by some other parameter.

S23. The window sequence numbers and interleave sequence numbers are recorded in the TARP headers of each packet.

S24. One TARP router address is randomly chosen for each TARP packet and the IP address corresponding to it stored for use in the clear IP header. The link key corresponding to this router is identified and used to encrypt TARP packets containing interleaved and encrypted data and TARP headers.

S25. A clear IP header with the first hop router's real IP address is generated and added to each of the encrypted TARP packets and the resulting packets.

Referring to FIG. **7**, the following particular steps may be employed in the above-described method for receiving TARP packets.

S40. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.

S42. The TARP packet may be probed to authenticate the packet before attempting to decrypt it using the link key.

S43. The TARP packet is decrypted with the appropriate link key to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.

S44. If the packet is a decoy packet, the perishable decoy counter is incremented.

S45. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the receiver may choose to throw it away.

S46. The TARP packets are cached until all packets forming an interleave window are received.

S47. Once all packets of an interleave window are received, the packets are deinterleaved.

S48. The packets block of combined packets defining the interleave window is then decrypted using the session key.

S49. The decrypted block is then divided using the window sequence data and the IP_T headers are converted into normal IP_C headers. The window sequence numbers are integrated in the IP_C headers.

S50. The packets are then handed up to the IP layer processes.

1. SCALABILITY ENHANCEMENTS

The IP agility feature described above relies on the ability to transmit IP address changes to all TARP routers. The embodiments including this feature will be referred to as "boutique" embodiments due to potential limitations in scaling these features up for a large network, such as the Internet. (The "boutique" embodiments would, however, be robust for use in smaller networks, such as small virtual private networks, for example). One problem with the boutique embodiments is that if IP address changes are to occur frequently, the message traffic required to update all routers sufficiently quickly creates a serious burden on the Internet when the TARP router and/or client population gets large. The bandwidth burden added to the networks, for example in ICMP packets, that would be used to update all the TARP routers could overwhelm the Internet for a large scale implementation that approached the scale of the Internet. In other words, the boutique system's scalability is limited.

A system can be constructed which trades some of the features of the above embodiments to provide the benefits of IP agility without the additional messaging burden. This is accomplished by IP address-hopping according to shared algorithms that govern IP addresses used between links participating in communications sessions between nodes such as TARP nodes. (Note that the IP hopping technique is also applicable to the boutique embodiment.) The IP agility feature discussed with respect to the boutique system can be modified so that it becomes decentralized under this scalable regime and governed by the above-described shared algorithm. Other features of the boutique system may be combined with this new type of IP-agility.

The new embodiment has the advantage of providing IP agility governed by a local algorithm and set of IP addresses exchanged by each communicating pair of nodes. This local governance is session-independent in that it may govern communications between a pair of nodes, irrespective of the session or end points being transferred between the directly communicating pair of nodes.

In the scalable embodiments, blocks of IP addresses are allocated to each node in the network. (This scalability will increase in the future, when Internet Protocol addresses are increased to 128-bit fields, vastly increasing the number of distinctly addressable nodes). Each node can thus use any of the IP addresses assigned to that node to communicate with other nodes in the network. Indeed, each pair of communicating nodes can use a plurality of source IP addresses and destination IP addresses for communicating with each other.

Each communicating pair of nodes in a chain participating in any session stores two blocks of IP addresses, called netblocks, and an algorithm and randomization seed for selecting, from each netblock, the next pair of source/destination IP addresses that will be used to transmit the next message. In other words, the algorithm governs the sequential selection of IP-address pairs, one sender and one receiver IP address, from each netblock. The combination of algorithm, seed, and netblock (IP address block) will be called a "hopblock." A router issues separate transmit and receive hopblocks to its clients. The send address and the receive address of the IP header of each outgoing packet sent by the client are filled with the send and receive IP addresses generated by the algorithm. The algorithm is "clocked" (indexed) by a counter so that each time a pair is used, the algorithm turns out a new transmit pair for the next packet to be sent.

The router's receive hopblock is identical to the client's transmit hopblock. The router uses the receive hopblock to predict what the send and receive IP address pair for the next expected packet from that client will be. Since packets can be received out of order, it is not possible for the router to predict with certainty what IP address pair will be on the next sequential packet. To account for this problem, the router generates a range of predictions encompassing the number of possible transmitted packet send/receive addresses, of which the next packet received could leap ahead. Thus, if there is a vanishingly small probability that a given packet will arrive at the router ahead of 5 packets transmitted by the client before the given packet, then the router can generate a series of 6 send/receive IP address pairs (or "hop window") to compare with the next received packet. When a packet is received, it is marked in the hop window as such, so that a second packet with the same IP address pair will be discarded. If an out-of-sequence packet does not arrive within a predetermined timeout period, it can be requested for retransmission or simply discarded from the receive table, depending upon the protocol in use for that communications session, or possibly by convention.

When the router receives the client's packet, it compares the send and receive IP addresses of the packet with the next N predicted send and receive IP address pairs and rejects the packet if it is not a member of this set. Received packets that do not have the predicted source/destination IP addresses falling with the window are rejected, thus thwarting possible hackers. (With the number of possible combinations, even a fairly large window would be hard to fall into at random.) If it is a member of this set, the router accepts the packet and processes it further. This link-based IP-hopping strategy, referred to as "IHOP," is a network element that stands on its own and is not necessarily accompanied by elements of the boutique system described above. If the routing agility feature described in connection with the boutique embodiment is combined with this link-based IP-hopping strategy, the router's next step would be to decrypt the TARP header to determine the destination TARP router for the packet and determine what should be the next hop for the packet. The TARP router would then forward the packet to a random TARP router or the destination TARP router with which the source TARP router has a link-based IP hopping communication established.

FIG. **8** shows how a client computer **801** and a TARP router **811** can establish a secure session. When client **801** seeks to establish an IHOP session with TARP router **811**, the client **801** sends "secure synchronization" request ("SSYN") packet **821** to the TARP router **811**. This SYN packet **821** contains the client's **801** authentication token, and may be sent to the router **811** in an encrypted format. The source and destination IP numbers on the packet **821** are the client's **801** current fixed IP address, and a "known" fixed IP address for the router **811**. (For security purposes, it may be desirable to reject any packets from outside of the local network that are destined for the router's known fixed IP address.) Upon receipt and validation of the client's **801** SSYN packet **821**, the router **811** responds by sending an encrypted "secure synchronization acknowledgment" ("SSYN ACK") **822** to the client **801**. This SSYN ACK **822** will contain the transmit and receive hopblocks that the client **801** will use when communicating with the TARP router **811**. The client **801** will acknowledge the TARP router's **811** response packet **822** by generating an encrypted SSYN ACK ACK packet **823** which will be sent from the client's **801** fixed IP address and to the TARP router's **811** known fixed IP address. The client **801** will simultaneously generate a SSYN ACK ACK packet; this SSYN ACK packet, referred to as the Secure Session Initiation (SSI) packet **824**, will be sent with the first {sender, receiver} IP pair in the client's transmit table **921** (FIG. **9**), as specified in the transmit hopblock provided by the TARP router **811** in the SSYN ACK packet **822**. The TARP router **811** will respond to the SSI packet **824** with an SSI ACK packet **825**, which will be sent with the first {sender, receiver} IP pair in the TARP router's transmit table **923**. Once these packets have been successfully exchanged, the secure communications session is established, and all further secure communications between the client **801** and the TARP router **811** will be conducted via this secure session, as long as synchronization is maintained. If synchronization is lost, then the client **801** and TARP router **802** may re-establish the secure session by the procedure outlined in FIG. **8** and described above.

While the secure session is active, both the client **901** and TARP router **911** (FIG. **9**) will maintain their respective transmit tables **921**, **923** and receive tables **922**, **924**, as provided by the TARP router during session synchronization **822**. It is important that the sequence of IP pairs in the client's transmit table **921** be identical to those in the TARP router's receive table **924**; similarly, the sequence of IP pairs in the client's receive table **922** must be identical to those in the router's transmit table **923**. This is required for the session synchronization to be maintained. The client **901** need maintain only one transmit table **921** and one receive table **922** during the course of the secure session. Each sequential packet sent by the client **901** will employ the next {send, receive} IP address pair in the transmit table, regardless of TCP or UDP session. The TARP router **911** will expect each packet arriving from the client **901** to bear the next IP address pair shown in its receive table.

Since packets can arrive out of order, however, the router **911** can maintain a "look ahead" buffer in its receive table, and will mark previously-received IP pairs as invalid for future packets; any future packet containing an IP pair that is in the look-ahead buffer but is marked as previously received will be discarded. Communications from the TARP router **911** to the client **901** are maintained in an identical manner; in particular, the router **911** will select the next IP address pair from its transmit table **923** when constructing a packet to send to the client **901**, and the client **901** will maintain a look-ahead buffer of expected IP pairs on packets that it is receiving. Each TARP router will maintain separate pairs of transmit and receive tables for each client that is currently engaged in a secure session with or through that TARP router.

While clients receive their hopblocks from the first server linking them to the Internet, routers exchange hopblocks. When a router establishes a link-based IP-hopping communication regime with another router, each router of the pair exchanges its transmit hopblock. The transmit hopblock of each router becomes the receive hopblock of the other router. The communication between routers is governed as described by the example of a client sending a packet to the first router.

While the above strategy works fine in the IP milieu, many local networks that are connected to the Internet are Ethernet systems. In Ethernet, the IP addresses of the destination devices must be translated into hardware addresses, and vice versa, using known processes (“address resolution protocol,” and “reverse address resolution protocol”). However, if the link-based IP-hopping strategy is employed, the correlation process would become explosive and burdensome. An alternative to the link-based IP hopping strategy may be employed within an Ethernet network. The solution is to provide that the node linking the Internet to the Ethernet (call it the border node) use the link-based IP-hopping communication regime to communicate with nodes outside the Ethernet LAN. Within the Ethernet LAN, each TARP node would have a single IP address which would be addressed in the conventional way. Instead of comparing the {sender, receiver} IP address pairs to authenticate a packet, the intra-LAN TARP node would use one of the IP header extension fields to do so. Thus, the border node uses an algorithm shared by the intra-LAN TARP node to generate a symbol that is stored in the free field in the IP header, and the intra-LAN TARP node generates a range of symbols based on its prediction of the next expected packet to be received from that particular source IP address. The packet is rejected if it does not fall into the set of predicted symbols (for example, numerical values) or is accepted if it does. Communications from the intra-LAN TARP node to the border node are accomplished in the same manner, though the algorithm will necessarily be different for security reasons. Thus, each of the communicating nodes will generate transmit and receive tables in a similar manner to that of FIG. **9**; the intra-LAN TARP nodes transmit table will be identical to the border node’s receive table, and the intra-LAN TARP node’s receive table will be identical to the border node’s transmit table.

The algorithm used for IP address-hopping can be any desired algorithm. For example, the algorithm can be a given pseudo-random number generator that generates numbers of the range covering the allowed IP addresses with a given seed. Alternatively, the session participants can assume a certain type of algorithm and specify simply a parameter for applying the algorithm. For example the assumed algorithm could be a particular pseudo-random number generator and the session participants could simply exchange seed values.

Note that there is no permanent physical distinction between the originating and destination terminal nodes. Either device at either end point can initiate a synchronization of the pair. Note also that the authentication/synchronization-request (and acknowledgment) and hopblock-exchange may all be served by a single message so that separate message exchanges may not be required.

As another extension to the stated architecture, multiple physical paths can be used by a client, in order to provide link redundancy and further thwart attempts at denial of service and traffic monitoring. As shown in FIG. **10**, for example, client **1001** can establish three simultaneous sessions with each of three TARP routers provided by different ISPs **1011**, **1012**, **1013**. As an example, the client **1001** can use three different telephone lines **1021**, **1022**, **1023** to connect to the ISPs, or two telephone lines and a cable modem, etc. In this scheme, transmitted packets will be sent in a random fashion among the different physical paths. This architecture provides a high degree of communications redundancy, with improved immunity from denial-of-service attacks and traffic monitoring.

2. FURTHER EXTENSIONS

The following describes various extensions to the techniques, systems, and methods described above. As described above, the security of communications occurring between computers in a computer network (such as the Internet, an Ethernet, or others) can be enhanced by using seemingly random source and destination Internet Protocol (IP) addresses for data packets transmitted over the network. This feature prevents eavesdroppers from determining which computers in the network are communicating with each other while permitting the two communicating computers to easily recognize whether a given received data packet is legitimate or not. In one embodiment of the above-described systems, an IP header extension field is used to authenticate incoming packets on an Ethernet.

Various extensions to the previously described techniques described herein include: (1) use of hopped hardware or “MAC” addresses in broadcast type network; (2) a self-synchronization technique that permits a computer to automatically regain synchronization with a sender; (3) synchronization algorithms that allow transmitting and receiving computers to quickly re-establish synchronization in the event of lost packets or other events; and (4) a fast-packet rejection mechanism for rejecting invalid packets. Any or all of these extensions can be combined with the features described above in any of various ways.

A. Hardware Address Hopping

Internet protocol-based communications techniques on a LAN—or across any dedicated physical medium—typically embed the IP packets within lower-level packets, often referred to as “frames.” As shown in FIG. **11**, for example, a first Ethernet frame **1150** comprises a frame header **1101** and two embedded IP packets IP**1** and IP**2**, while a second Ethernet frame **1160** comprises a different frame header **1104** and a single IP packet IP**3**. Each frame header generally includes a source hardware address **1101**A and a destination hardware address **1101**B; other well-known fields in frame headers are omitted from FIG. **11** for clarity. Two hardware nodes communicating over a physical communication channel insert appropriate source and destination hardware addresses to indicate which nodes on the channel or network should receive the frame.

It may be possible for a nefarious listener to acquire information about the contents of a frame and/or its communicants by examining frames on a local network rather than (or in addition to) the IP packets themselves. This is especially true in broadcast media, such as Ethernet, where it is necessary to insert into the frame header the hardware address of the machine that generated the frame and the hardware address of the machine to which frame is being sent. All nodes on the network can potentially “see” all packets transmitted across the network. This can be a problem for secure communications, especially in cases where the communicants do not want for any third party to be able to identify who is engaging in the information exchange. One way to address this problem is to push the address-hopping scheme down to the hardware layer. In accordance with various embodiments of the invention, hardware addresses are “hopped” in a manner similar to that used to change IP addresses, such that a listener cannot determine which hardware node generated a particular message nor which node is the intended recipient.

FIG. **12**A shows a system in which Media Access Control ("MAC") hardware addresses are "hopped" in order to increase security over a network such as an Ethernet. While the description refers to the exemplary case of an Ethernet environment, the inventive principles are equally applicable to other types of communications media. In the Ethernet case, the MAC address of the sender and receiver are inserted into the Ethernet frame and can be observed by anyone on the LAN who is within the broadcast range for that frame. For secure communications, it becomes desirable to generate frames with MAC addresses that are not attributable to any specific sender or receiver.

As shown in FIG. **12**A, two computer nodes **1201** and **1202** communicate over a communication channel such as an Ethernet. Each node executes one or more application programs **1203** and **1218** that communicate by transmitting packets through communication software **1204** and **1217**, respectively. Examples of application programs include video conferencing, e-mail, word processing programs, telephony, and the like. Communication software **1204** and **1217** can comprise, for example, an OSI layered architecture or "stack" that standardizes various services provided at different levels of functionality.

The lowest levels of communication software **1204** and **1217** communicate with hardware components **1206** and **1214** respectively, each of which can include one or more registers **1207** and **1215** that allow the hardware to be reconfigured or controlled in accordance with various communication protocols. The hardware components (an Ethernet network interface card, for example) communicate with each other over the communication medium. Each hardware component is typically pre-assigned a fixed hardware address or MAC number that identifies the hardware component to other nodes on the network. One or more interface drivers control the operation of each card and can, for example, be configured to accept or reject packets from certain hardware addresses. As will be described in more detail below, various embodiments of the inventive principles provide for "hopping" different addresses using one or more algorithms and one or more moving windows that track a range of valid addresses to validate received packets. Packets transmitted according to one or more of the inventive principles will be generally referred to as "secure" packets or "secure communications" to differentiate them from ordinary data packets that are transmitted in the clear using ordinary, machine-correlated addresses.

One straightforward method of generating non-attributable MAC addresses is an extension of the IP hopping scheme. In this scenario, two machines on the same LAN that desire to communicate in a secure fashion exchange random-number generators and seeds, and create sequences of quasi-random MAC addresses for synchronized hopping. The implementation and synchronization issues are then similar to that of IP hopping.

This approach, however, runs the risk of using MAC addresses that are currently active on the LAN-which, in turn, could interrupt communications for those machines. Since an Ethernet MAC address is at present 48 bits in length, the chance of randomly misusing an active MAC address is actually quite small. However, if that figure is multiplied by a large number of nodes (as would be found on an extensive LAN), by a large number of frames (as might be the case with packet voice or streaming video), and by a large number of concurrent Virtual Private Networks (VPNs), then the chance that a non-secure machine's MAC address could be used in an

address-hopped frame can become non-trivial. In short, any scheme that runs even a small risk of interrupting communications for other machines on the LAN is bound to receive resistance from prospective system administrators. Nevertheless, it is technically feasible, and can be implemented without risk on a LAN on which there is a small number of machines, or if all of the machines on the LAN are engaging in MAC-hopped communications.

Synchronized MAC address hopping may incur some overhead in the course of session establishment, especially if there are multiple sessions or multiple nodes involved in the communications. A simpler method of randomizing MAC addresses is to allow each node to receive and process every incident frame on the network. Typically, each network interface driver will check the destination MAC address in the header of every incident frame to see if it matches that machine's MAC address; if there is no match, then the frame is discarded. In one embodiment, however, these checks can be disabled, and every incident packet is passed to the TARP stack for processing. This will be referred to as "promiscuous" mode, since every incident frame is processed. Promiscuous mode allows the sender to use completely random, unsynchronized MAC addresses, since the destination machine is guaranteed to process the frame. The decision as to whether the packet was truly intended for that machine is handled by the TARP stack, which checks the source and destination IP addresses for a match in its IP synchronization tables. If no match is found, the packet is discarded; if there is a match, the packet is unwrapped, the inner header is evaluated, and if the inner header indicates that the packet is destined for that machine then the packet is forwarded to the IP stack—otherwise it is discarded.

One disadvantage of purely-random MAC address hopping is its impact on processing overhead; that is, since every incident frame must be processed, the machine's CPU is engaged considerably more often than if the network interface driver is discriminating and rejecting packets unilaterally. A compromise approach is to select either a single fixed MAC address or a small number of MAC addresses (e.g., one for each virtual private network on an Ethernet) to use for MAC-hopped communications, regardless of the actual recipient for which the message is intended. In this mode, the network interface driver can check each incident frame against one (or a few) pre-established MAC addresses, thereby freeing the CPU from the task of physical-layer packet discrimination. This scheme does not betray any useful information to an interloper on the LAN; in particular, every secure packet can already be identified by a unique packet type in the outer header. However, since all machines engaged in secure communications would either be using the same MAC address, or be selecting from a small pool of predetermined MAC addresses, the association between a specific machine and a specific MAC address is effectively broken.

In this scheme, the CPU will be engaged more often than it would be in non-secure communications (or in synchronized MAC address hopping), since the network interface driver cannot always unilaterally discriminate between secure packets that are destined for that machine, and secure packets from other VPNs. However, the non-secure traffic is easily eliminated at the network interface, thereby reducing the amount of processing required of the CPU. There are boundary conditions where these statements would not hold, of course—e.g., if all of the traffic on the LAN is secure traffic, then the CPU would be engaged to the same degree as it is in the purely-random address hopping case; alternatively, if each VPN on the LAN uses a different MAC address, then the network interface can perfectly discriminate secure frames destined for the local machine from those constituting other VPNs. These are engineering tradeoffs that might be best handled by providing administrative options for the users when installing the software and/or establishing VPNs.

Even in this scenario, however, there still remains a slight risk of selecting MAC addresses that are being used by one or more nodes on the LAN. One solution to this problem is to formally assign one address or a range of addresses for use in MAC-hopped communications. This is typically done via an assigned numbers registration authority; e.g., in the case of Ethernet, MAC address ranges are assigned to vendors by the Institute of Electrical and Electronics Engineers (IEEE). A formally-assigned range of addresses would ensure that secure frames do not conflict with any properly-configured and properly-functioning machines on the LAN.

Reference will now be made to FIGS. **12**A and **12**B in order to describe the many combinations and features that follow the inventive principles. As explained above, two computer nodes **1201** and **1202** are assumed to be communicating over a network or communication medium such as an Ethernet. A communication protocol in each node (**1204** and **1217**, respectively) contains a modified element **1205** and **1216** that performs certain functions that deviate from the standard communication protocols. In particular, computer node **1201** implements a first "hop" algorithm **1208**X that selects seemingly random source and destination IP addresses (and, in one embodiment, seemingly random IP header discriminator fields) in order to transmit each packet to the other computer node. For example, node **1201** maintains a transmit table **1208** containing triplets of source (S), destination (D), and discriminator fields (DS) that are inserted into outgoing IP packet headers. The table is generated through the use of an appropriate algorithm (e.g., a random number generator that is seeded with an appropriate seed) that is known to the recipient node **1202**. As each new IP packet is formed, the next sequential entry out of the sender's transmit table **1208** is used to populate the IP source, IP destination, and IP header extension field (e.g., discriminator field). It will be appreciated that the transmit table need not be created in advance but could instead be created on-the-fly by executing the algorithm when each packet is formed.

At the receiving node **1202**, the same IP hop algorithm **1222**X is maintained and used to generate a receive table **1222** that lists valid triplets of source IP address, destination IP address, and discriminator field. This is shown by virtue of the first five entries of transmit table **1208** matching the second five entries of receive table **1222**. (The tables may be slightly offset at any particular time due to lost packets, misordered packets, or transmission delays). Additionally, node **1202** maintains a receive window W**3** that represents a list of valid IP source, IP destination, and discriminator fields that will be accepted when received as part of an incoming IP packet. As packets are received, window W**3** slides down the list of valid entries, such that the possible valid entries change over time. Two packets that arrive out of order but are nevertheless matched to entries within window W**3** will be accepted; those falling outside of window W**3** will be rejected as invalid. The length of window W**3** can be adjusted as necessary to reflect network delays or other factors.

Node **1202** maintains a similar transmit table **1221** for creating IP packets and frames destined for node **1201** using a potentially different hopping algorithm **1221**X, and node **1201** maintains a matching receive table **1209** using the same algorithm **1209**X. As node **1202** transmits packets to node **1201** using seemingly random IP source, IP destination, and/or discriminator fields, node **1201** matches the incoming packet values to those falling within window W**1** maintained in its receive table. In effect, transmit table **1208** of node **1201** is synchronized (i.e., entries are selected in the same order) to receive table **1222** of receiving node **1202**. Similarly, transmit table **1221** of node **1202** is synchronized to receive table **1209** of node **1201**. It will be appreciated that although a common algorithm is shown for the source, destination and discriminator fields in FIG. **12**A (using, e.g., a different seed for each of the three fields), an entirely different algorithm could in fact be used to establish values for each of these fields. It will also be appreciated that one or two of the fields can be "hopped" rather than all three as illustrated.

In accordance with another aspect of the invention, hardware or "MAC" addresses are hopped instead of or in addition to IP addresses and/or the discriminator field in order to improve security in a local area or broadcast-type network. To that end, node **1201** further maintains a transmit table **1210** using a transmit algorithm **1210**X to generate source and destination hardware addresses that are inserted into frame headers (e.g., fields **1101**A and **1101**B in FIG. **11**) that are synchronized to a corresponding receive table **1224** at node **1202**. Similarly, node **1202** maintains a different transmit table **1223** containing source and destination hardware addresses that is synchronized with a corresponding receive table **1211** at node **1201**. In this manner, outgoing hardware frames appear to be originating from and going to completely random nodes on the network, even though each recipient can determine whether a given packet is intended for it or not. It will be appreciated that the hardware hopping feature can be implemented at a different level in the communications protocol than the IP hopping feature (e.g., in a card driver or in a hardware card itself to improve performance).

FIG. **12**B shows three different embodiments or modes that can be employed using the aforementioned principles. In a first mode referred to as "promiscuous" mode, a common hardware address (e.g., a fixed address for source and another for destination) or else a completely random hardware address is used by all nodes on the network, such that a particular packet cannot be attributed to any one node. Each node must initially accept all packets containing the common (or random) hardware address and inspect the IP addresses or discriminator field to determine whether the packet is intended for that node. In this regard, either the IP addresses or the discriminator field or both can be varied in accordance with an algorithm as described above. As explained previously, this may increase each node's overhead since additional processing is involved to determine whether a given packet has valid source and destination hardware addresses.

In a second mode referred to as "promiscuous per VPN" mode, a small set of fixed hardware addresses are used, with a fixed source/destination hardware address used for all nodes communicating over a virtual private network. For example, if there are six nodes on an Ethernet, and the network is to be split up into two private virtual networks such that nodes on one VPN can communicate with only the other two nodes on its own VPN, then two sets of hardware addresses could be used: one set for the first VPN and a second set for the second VPN. This would reduce the amount of overhead involved in checking for valid frames since only packets arriving from the designated VPN would need to be checked. IP addresses and one or more discriminator fields could still be hopped as before for secure communication within the VPN. Of course, this solution compromises the anonymity of the VPNs (i.e., an outsider can easily tell what traffic belongs in which VPN, though he cannot correlate it to a specific machine/person). It also requires the use of a discriminator field to mitigate the vulnerability to certain types of DoS attacks. (For example, without the discriminator field, an attacker on the LAN could stream frames containing the MAC addresses being used by the VPN; rejecting those frames could lead to excessive processing overhead. The discriminator field would provide a low-overhead means of rejecting the false packets.)

In a third mode referred to as "hardware hopping" mode, hardware addresses are varied as illustrated in FIG. **12**A, such that hardware source and destination addresses are changed constantly in order to provide non-attributable addressing. Variations on these embodiments are of course possible, and the invention is not intended to be limited in any respect by these illustrative examples.

B. Extending the Address Space

Address hopping provides security and privacy. However, the level of protection is limited by the number of addresses in the blocks being hopped. A hopblock denotes a field or fields modulated on a packet-wise basis for the purpose of providing a VPN. For instance, if two nodes communicate with IP address hopping using hopblocks of 4 addresses (2 bits) each, there would be 16 possible address-pair combinations. A window of size 16 would result in most address pairs being accepted as valid most of the time. This limitation can be overcome by using a discriminator field in addition to or instead of the hopped address fields. The discriminator field would be hopped in exactly the same fashion as the address fields and it would be used to determine whether a packet should be processed by a receiver.

Suppose that two clients, each using four-bit hopblocks, would like the same level of protection afforded to clients communicating via IP hopping between two A blocks (24 address bits eligible for hopping). A discriminator field of 20 bits, used in conjunction with the 4 address bits eligible for hopping in the IP address field, provides this level of protection. A 24-bit discriminator field would provide a similar level of protection if the address fields were not hopped or ignored. Using a discriminator field offers the following advantages: (1) an arbitrarily high level of protection can be provided, and (2) address hopping is unnecessary to provide protection. This may be important in environments where address hopping would cause routing problems.

C. Synchronization Techniques

It is generally assumed that once a sending node and receiving node have exchanged algorithms and seeds (or similar information sufficient to generate quasi-random source and destination tables), subsequent communication between the two nodes will proceed smoothly. Realistically, however, two nodes may lose synchronization due to network delays or outages, or other problems. Consequently, it is desirable to provide means for re-establishing synchronization between nodes in a network that have lost synchronization.

One possible technique is to require that each node provide an acknowledgment upon successful receipt of each packet and, if no acknowledgment is received within a certain period of time, to re-send the unacknowledged packet. This approach, however, drives up overhead costs and may be prohibitive in high-throughput environments such as streaming video or audio, for example.

A different approach is to employ an automatic synchronizing technique that will be referred to herein as "self-synchronization." In this approach, synchronization information is embedded into each packet, thereby enabling the receiver to re-synchronize itself upon receipt of a single packet if it determines that is has lost synchronization with the sender. (If communications are already in progress, and the receiver determines that it is still in sync with the sender, then there is no need to re-synchronize.) A receiver could detect that it was out of synchronization by, for example, employing a "dead-man" timer that expires after a certain period of time, wherein the timer is reset with each valid packet. A time stamp could be hashed into the public sync field (see below) to preclude packet-retry attacks.

In one embodiment, a "sync field" is added to the header of each packet sent out by the sender. This sync field could appear in the clear or as part of an encrypted portion of the packet. Assuming that a sender and receiver have selected a random-number generator (RNG) and seed value, this combination of RNG and seed can be used to generate a random-number sequence (RNS). The RNS is then used to generate a sequence of source/destination IP pairs (and, if desired, discriminator fields and hardware source and destination addresses), as described above. It is not necessary, however, to generate the entire sequence (or the first N−1 values) in order to generate the Nth random number in the sequence; if the sequence index N is known, the random value corresponding to that index can be directly generated (see below). Different RNGs (and seeds) with different fundamental periods could be used to generate the source and destination IP sequences, but the basic concepts would still apply. For the sake of simplicity, the following discussion will assume that IP source and destination address pairs (only) are hopped using a single RNG sequencing mechanism.

In accordance with a "self-synchronization" feature, a sync field in each packet header provides an index (i.e., a sequence number) into the RNS that is being used to generate IP pairs. Plugging this index into the RNG that is being used to generate the RNS yields a specific random number value, which in turn yields a specific IP pair. That is, an IP pair can be generated directly from knowledge of the RNG, seed, and index number; it is not necessary, in this scheme, to generate the entire sequence of random numbers that precede the sequence value associated with the index number provided.

Since the communicants have presumably previously exchanged RNGs and seeds, the only new information that must be provided in order to generate an IP pair is the sequence number. If this number is provided by the sender in the packet header, then the receiver need only plug this number into the RNG in order to generate an IP pair—and thus verify that the IP pair appearing in the header of the packet is valid. In this scheme, if the sender and receiver lose synchronization, the receiver can immediately re-synchronize upon receipt of a single packet by simply comparing the IP pair in the packet header to the IP pair generated from the index number. Thus, synchronized communications can be resumed upon receipt of a single packet, making this scheme ideal for multicast communications. Taken to the extreme, it could obviate the need for synchronization tables entirely; that is, the sender and receiver could simply rely on the index number in the sync field to validate the IP pair on each packet, and thereby eliminate the tables entirely.

The aforementioned scheme may have some inherent security issues associated with it—namely, the placement of the sync field. If the field is placed in the outer header, then an interloper could observe the values of the field and their relationship to the IP stream. This could potentially compromise the algorithm that is being used to generate the IP-address sequence, which would compromise the security of the communications. If, however, the value is placed in the inner header, then the sender must decrypt the inner header before it can extract the sync value and validate the IP pair;

this opens up the receiver to certain types of denial-of-service (DoS) attacks, such as packet replay. That is, if the receiver must decrypt a packet before it can validate the IP pair, then it could potentially be forced to expend a significant amount of processing on decryption if an attacker simply retransmits previously valid packets. Other attack methodologies are possible in this scenario.

A possible compromise between algorithm security and processing speed is to split up the sync value between an inner (encrypted) and outer (unencrypted) header. That is, if the sync value is sufficiently long, it could potentially be split into a rapidly-changing part that can be viewed in the clear, and a fixed (or very slowly changing) part that must be protected. The part that can be viewed in the clear will be called the "public sync" portion and the part that must be protected will be called the "private sync" portion.

Both the public sync and private sync portions are needed to generate the complete sync value. The private portion, however, can be selected such that it is fixed or will change only occasionally. Thus, the private sync value can be stored by the recipient, thereby obviating the need to decrypt the header in order to retrieve it. If the sender and receiver have previously agreed upon the frequency with which the private part of the sync will change, then the receiver can selectively decrypt a single header in order to extract the new private sync if the communications gap that has led to lost synchronization has exceeded the lifetime of the previous private sync. This should not represent a burdensome amount of decryption, and thus should not open up the receiver to denial-of-service attack simply based on the need to occasionally decrypt a single header.

One implementation of this is to use a hashing function with a one-to-one mapping to generate the private and public sync portions from the sync value. This implementation is shown in FIG. **13**, where (for example) a first ISP **1302** is the sender and a second ISP **1303** is the receiver. (Other alternatives are possible from FIG. **13**.) A transmitted packet comprises a public or "outer" header **1305** that is not encrypted, and a private or "inner" header **1306** that is encrypted using for example a link key. Outer header **1305** includes a public sync portion while inner header **1306** contains the private sync portion. A receiving node decrypts the inner header using a decryption function **1307** in order to extract the private sync portion. This step is necessary only if the lifetime of the currently buffered private sync has expired. (If the currently-buffered private sync is still valid, then it is simply extracted from memory and "added" (which could be an inverse hash) to the public sync, as shown in step **1308**.) The public and decrypted private sync portions are combined in function **1308** in order to generate the combined sync **1309**. The combined sync (**1309**) is then fed into the RNG (**1310**) and compared to the IP address pair (**1311**) to validate or reject the packet.

An important consideration in this architecture is the concept of "future" and "past" where the public sync values are concerned. Though the sync values, themselves, should be random to prevent spoofing attacks, it may be important that the receiver be able to quickly identify a sync value that has already been sent—even if the packet containing that sync value was never actually received by the receiver. One solution is to hash a time stamp or sequence number into the public sync portion, which could be quickly extracted, checked, and discarded, thereby validating the public sync portion itself.

In one embodiment, packets can be checked by comparing the source/destination IP pair generated by the sync field with the pair appearing in the packet header. If (1) they match, (2) the time stamp is valid, and (3) the dead-man timer has expired, then re-synchronization occurs; otherwise, the packet is rejected. If enough processing power is available, the dead-man timer and synchronization tables can be avoided altogether, and the receiver would simply resynchronize (e.g., validate) on every packet.

The foregoing scheme may require large-integer (e.g., 160-bit) math, which may affect its implementation. Without such large-integer registers, processing throughput would be affected, thus potentially affecting security from a denial-of-service standpoint. Nevertheless, as large-integer math processing features become more prevalent, the costs of implementing such a feature will be reduced.

D. Other Synchronization Schemes

As explained above, if W or more consecutive packets are lost between a transmitter and receiver in a VPN (where W is the window size), the receiver's window will not have been updated and the transmitter will be transmitting packets not in the receiver's window. The sender and receiver will not recover synchronization until perhaps the random pairs in the window are repeated by chance. Therefore, there is a need to keep a transmitter and receiver in synchronization whenever possible and to re-establish synchronization whenever it is lost.

A "checkpoint" scheme can be used to regain synchronization between a sender and a receiver that have fallen out of synchronization. In this scheme, a checkpoint message comprising a random IP address pair is used for communicating synchronization information. In one embodiment, two messages are used to communicate synchronization information between a sender and a recipient:

1. SYNC_REQ is a message used by the sender to indicate that it wants to synchronize; and
2. SYNC_ACK is a message used by the receiver to inform the transmitter that it has been synchronized.

According to one variation of this approach, both the transmitter and receiver maintain three checkpoints (see FIG. **14**):

1. In the transmitter, ckpt_o ("checkpoint old") is the IP pair that was used to re-send the last SYNC_REQ packet to the receiver. In the receiver, ckpt_o ("checkpoint old") is the IP pair that receives repeated SYNC_REQ packets from the transmitter.
2. In the transmitter, ckpt_n ("checkpoint new") is the IP pair that will be used to send the next SYNC_REQ packet to the receiver. In the receiver, ckpt_n ("checkpoint new") is the IP pair that receives a new SYNC_REQ packet from the transmitter and which causes the receiver's window to be re-aligned, ckpt_o set to ckpt_n, a new ckpt_n to be generated and a new ckpt_r to be generated.
3. In the transmitter, ckpt_r is the IP pair that will be used to send the next SYNC_ACK packet to the receiver. In the receiver, ckpt_r is the IP pair that receives a new SYNC_ACK packet from the transmitter and which causes a new ckpt_n to be generated. Since SYNC_ACK is transmitted from the receiver ISP to the sender ISP, the transmitter ckpt_r refers to the ckpt_r of the receiver and the receiver ckpt_r refers to the ckpt_r of the transmitter (see FIG. **14**).

When a transmitter initiates synchronization, the IP pair it will use to transmit the next data packet is set to a predetermined value and when a receiver first receives a SYNC_REQ, the receiver window is updated to be centered on the transmitter's next IP pair. This is the primary mechanism for checkpoint synchronization.

Synchronization can be initiated by a packet counter (e.g., after every N packets transmitted, initiate a synchronization) or by a timer (every S seconds, initiate a synchronization) or a combination of both. See FIG. **15**. From the transmitter's perspective, this technique operates as follows: (1) Each transmitter periodically transmits a "sync request" message to the receiver to make sure that it is in sync. (2) If the receiver is still in sync, it sends back a "sync ack" message. (If this works, no further action is necessary). (3) If no "sync ack" has been received within a period of time, the transmitter retransmits the sync request again. If the transmitter reaches the next checkpoint without receiving a "sync ack" response, then synchronization is broken, and the transmitter should stop transmitting. The transmitter will continue to send sync_reqs until it receives a sync_ack, at which point transmission is reestablished.

From the receiver's perspective, the scheme operates as follows: (1) when it receives a "sync request" request from the transmitter, it advances its window to the next checkpoint position (even skipping pairs if necessary), and sends a "sync ack" message to the transmitter. If sync was never lost, then the "jump ahead" really just advances to the next available pair of addresses in the table (i.e., normal advancement).

If an interloper intercepts the "sync request" messages and tries to interfere with communication by sending new ones, it will be ignored if the synchronization has been established or it it will actually help to re-establish synchronization.

A window is realigned whenever a re-synchronization occurs. This realignment entails updating the receiver's window to straddle the address pairs used by the packet transmitted immediately after the transmission of the SYNC_REQ packet. Normally, the transmitter and receiver are in synchronization with one another. However, when network events occur, the receiver's window may have to be advanced by many steps during resynchronization. In this case, it is desirable to move the window ahead without having to step through the intervening random numbers sequentially. (This feature is also desirable for the auto-sync approach discussed above).

E. Random Number Generator with a Jump-Ahead capability

An attractive method for generating randomly hopped addresses is to use identical random number generators in the transmitter and receiver and advance them as packets are transmitted and received. There are many random number generation algorithms that could be used. Each one has strengths and weaknesses for address hopping applications.

Linear congruential random number generators (LCRs) are fast, simple and well characterized random number generators that can be made to jump ahead n steps efficiently. An LCR generates random numbers X_1, X_2, X_3 . . . X_k starting with seed X_0 using a recurrence

$$X_i=(a\,X_{i-1}+b) \bmod c \qquad (1)$$

where a, b and c define a particular LCR. Another expression for X_i,

$$X_i=((a^i(X_0+b)-b)/(a-1)) \bmod c \qquad (2)$$

enables the jump-ahead capability. The factor a^i can grow very large even for modest i if left unfettered. Therefore some special properties of the modulo operation can be used to control the size and processing time required to compute (2). (2) can be rewritten as:

$$X_i=(a^i(X_0(a-1)+b)-b)/(a-1) \bmod c \qquad (3)$$

It can be shown that:

$$(a^i(X_0(a-1)+b)-b)/(a-1) \bmod c=((a^i \bmod((a-1)frxc)(X_0(a-1)+b)-b)/(a-1)) \bmod c \qquad (4)$$

$(X_0(a-1)+b)$ can be stored as $(X_0(a-1)+b) \bmod c$, b as b mod c and compute a^i mod ((a−1)c)(this requires O(log(i)) steps).

A practical implementation of this algorithm would jump a fixed distance, n, between synchronizations; this is tantamount to synchronizing every n packets. The window would commence n IP pairs from the start of the previous window. Using X_j^w, the random number at the j^{th} checkpoint, as X_0 and n as i, a node can store a^n mod((a−1)c) once per LCR and set

$$X_{j+1}^w=X_{n(j+1)}=((a^n \bmod((a+1)c)(X_j^w(a-1)+b)-b)/(a-1)) \bmod c, \qquad (5)$$

to generate the random number for the $j+1^{th}$ synchronization. Using this construction, a node could jump ahead an arbitrary (but fixed) distance between synchronizations in a constant amount of time (independent of n).

Pseudo-random number generators, in general, and LCRs, in particular, will eventually repeat their cycles. This repetition may present vulnerability in the IP hopping scheme. An adversary would simply have to wait for a repeat to predict future sequences. One way of coping with this vulnerability is to create a random number generator with a known long cycle. A random sequence can be replaced by a new random number generator before it repeats. LCRs can be constructed with known long cycles. This is not currently true of many random number generators.

Random number generators can be cryptographically insecure. An adversary can derive the RNG parameters by examining the output or part of the output. This is true of LCGs. This vulnerability can be mitigated by incorporating an encryptor, designed to scramble the output as part of the random number generator. The random number generator prevents an adversary from mounting an attack—e.g., a known plaintext attack—against the encryptor.

F. Random Number Generator Example

Consider a RNG where a=31, b=4 and c=15. For this case equation (1) becomes:

$$X_i=(31\,X_{i-1}+4) \bmod 15. \qquad (6)$$

If one sets X_0=1, equation (6) will produce the sequence 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 0, 4, 8, 12. This sequence will repeat indefinitely. For a jump ahead of 3 numbers in this sequence $a^n=31^3=29791$, c*(a−1)=15*30=450 and a^n mod ((a−1)c)=31^3mod(15*30)=29791mod(450)=91. Equation (5) becomes:

$$(91(X_i30+4)-4)/30) \bmod 15 \qquad (7)$$

Table 1 shows the jump ahead calculations from (7). The calculations start at 5 and jump ahead 3.

TABLE 1

I	X_i	(X_i30 + 4)	91 (X_i30 + 4) − 4	((91 (X_i30 + 4) − 4) / 30	X_{i+3}
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

G. Fast Packet Filter

Address hopping VPNs must rapidly determine whether a packet has a valid header and thus requires further processing, or has an invalid header (a hostile packet) and should be immediately rejected. Such rapid determinations will be referred to as "fast packet filtering." This capability protects the VPN from attacks by an adversary who streams hostile packets at the receiver at a high rate of speed in the hope of saturating the receiver's processor (a so-called "denial of service" attack). Fast packet filtering is an important feature for implementing VPNs on shared media such as Ethernet.

Assuming that all participants in a VPN share an unassigned "A" block of addresses, one possibility is to use an experimental "A" block that will never be assigned to any machine that is not address hopping on the shared medium. "A" blocks have a 24 bits of address that can be hopped as opposed to the 8 bits in "C" blocks. In this case a hopblock will be the "A" block. The use of the experimental "A" block is a likely option on an Ethernet because:

1. The addresses have no validity outside of the Ethernet and will not be routed out to a valid outside destination by a gateway.
2. There are 2^{24} (~16 million) addresses that can be hopped within each "A" block. This yields >280 trillion possible address pairs making it very unlikely that an adversary would guess a valid address. It also provides acceptably low probability of collision between separate VPNs (all VPNs on a shared medium independently generate random address pairs from the same "A" block).
3. The packets will not be received by someone on the Ethernet who is not on a VPN (unless the machine is in promiscuous mode) minimizing impact on non-VPN computers.

The Ethernet example will be used to describe one implementation of fast packet filtering. The ideal algorithm would quickly examine a packet header, determine whether the packet is hostile, and reject any hostile packets or determine which active IP pair the packet header matches. The problem is a classical associative memory problem. A variety of techniques have been developed to solve this problem (hashing, B-trees etc). Each of these approaches has its strengths and weaknesses. For instance, hash tables can be made to operate quite fast in a statistical sense, but can occasionally degenerate into a much slower algorithm. This slowness can persist for a period of time. Since there is a need to discard hostile packets quickly at all times, hashing would be unacceptable.

H. Presence Vector Alorithm

A presence vector is a bit vector of length 2^n that can be indexed by n-bit numbers (each ranging from 0 to 2^n-1). One can indicate the presence of k n-bit numbers (not necessarily unique), by setting the bits in the presence vector indexed by each number to 1. Otherwise, the bits in the presence vector are 0. An n-bit number, x, is one of the k numbers if and only if the x^{th} bit of the presence vector is 1. A fast packet filter can be implemented by indexing the presence vector and looking for a 1, which will be referred to as the "test."

For example, suppose one wanted to represent the number 135 using a presence vector. The 135^{th} bit of the vector would be set. Consequently, one could very quickly determine whether an address of 135 was valid by checking only one bit: the 135^{th} bit. The presence vectors could be created in advance corresponding to the table entries for the IP addresses. In effect, the incoming addresses can be used as indices into a long vector, making comparisons very fast. As each RNG generates a new address, the presence vector is updated to reflect the information. As the window moves, the presence vector is updated to zero out addresses that are no longer valid.

There is a trade-off between efficiency of the test and the amount of memory required for storing the presence vector(s). For instance, if one were to use the 48 bits of hopping addresses as an index, the presence vector would have to be 35 terabytes. Clearly, this is too large for practical purposes. Instead, the 48 bits can be divided into several smaller fields. For instance, one could subdivide the 48 bits into four 12-bit fields (see FIG. **16**). This reduces the storage requirement to 2048 bytes at the expense of occasionally having to process a hostile packet. In effect, instead of one long presence vector, the decomposed address portions must match all four shorter presence vectors before further processing is allowed. (If the first part of the address portion doesn't match the first presence vector, there is no need to check the remaining three presence vectors).

A presence vector will have a 1 in the y^{th} bit if and only if one or more addresses with a corresponding field of y are active. An address is active only if each presence vector indexed by the appropriate sub-field of the address is 1.

Consider a window of 32 active addresses and 3 checkpoints. A hostile packet will be rejected by the indexing of one presence vector more than 99% of the time. A hostile packet will be rejected by the indexing of all 4 presence vectors more than 99.9999995% of the time. On average, hostile packets will be rejected in less than 1.02 presence vector index operations.

The small percentage of hostile packets that pass the fast packet filter will be rejected when matching pairs are not found in the active window or are active checkpoints. Hostile packets that serendipitously match a header will be rejected when the VPN software attempts to decrypt the header. However, these cases will be extremely rare. There are many other ways this method can be configured to arbitrate the space/speed tradeoffs.

I. Further Synchronization Enhancements

A slightly modified form of the synchronization techniques described above can be employed. The basic principles of the previously described checkpoint synchronization scheme remain unchanged. The actions resulting from the reception of the checkpoints are, however, slightly different. In this variation, the receiver will maintain between OoO ("Out of Order") and 2×WINDOW_SIZE+OoO active addresses (1≦OoO≦WINDOW_SIZE and WINDOW_SIZE≧1). OoO and WINDOW_SIZE are engineerable parameters, where OoO is the minimum number of addresses needed to accommodate lost packets due to events in the network or out of order arrivals and WINDOW_SIZE is the number of packets transmitted before a SYNC_REQ is issued. FIG. **17** depicts a storage array for a receiver's active addresses.

The receiver starts with the first 2×WINDOW_SIZE addresses loaded and active (ready to receive data). As packets are received, the corresponding entries are marked as "used" and are no longer eligible to receive packets. The transmitter maintains a packet counter, initially set to 0, containing the number of data packets transmitted since the last initial transmission of a SYNC_REQ for which SYNC_ACK has been received. When the transmitter packet counter equals WINDOW_SIZE, the transmitter generates a SYNC_REQ and does its initial transmission. When the receiver receives a SYNC_REQ corresponding to its current CKPT_N, it generates the next WINDOW_SIZE addresses and starts loading them in order starting at the first location after the last active address wrapping around to the beginning of the array after the end of the array has been reached. The receiver's array might look like FIG. **18** when a SYNC_REQ has been received. In this case a couple of packets have been either lost or will be received out of order when the SYNC_REQ is received.

FIG. **19** shows the receiver's array after the new addresses have been generated. If the transmitter does not receive a SYNC_ACK, it will re-issue the SYNC_REQ at regular intervals. When the transmitter receives a SYNC_ACK, the packet counter is decremented by WINDOW_SIZE. If the packet counter reaches 2×WINDOW_SIZE –OoO then the transmitter ceases sending data packets until the appropriate SYNC_ACK is finally received. The transmitter then resumes sending data packets. Future behavior is essentially a repetition of this initial cycle. The advantages of this approach are:

1. There is no need for an efficient jump ahead in the random number generator,
2. No packet is ever transmitted that does not have a corresponding entry in the receiver side
3. No timer based re-synchronization is necessary. This is a consequence of 2.
4. The receiver will always have the ability to accept data messages transmitted within OoO messages of the most recently transmitted message.

J. Distributed Transmission Path Variant

Another embodiment incorporating various inventive principles is shown in FIG. **20**. In this embodiment, a message transmission system includes a first computer **2001** in communication with a second computer **2002** through a network **2011** of intermediary computers. In one variant of this embodiment, the network includes two edge routers **2003** and **2004** each of which is linked to a plurality of Internet Service Providers (ISPs) **2005** through **2010**. Each ISP is coupled to a plurality of other ISPs in an arrangement as shown in FIG. **20**, which is a representative configuration only and is not intended to be limiting. Each connection between ISPs is labeled in FIG. **20** to indicate a specific physical transmission path (e.g., AD is a physical path that links ISP A (element **2005**) to ISP D (element **2008**)). Packets arriving at each edge router are selectively transmitted to one of the ISPs to which the router is attached on the basis of a randomly or quasi-randomly selected basis.

As shown in FIG. **21**, computer **2001** or edge router **2003** incorporates a plurality of link transmission tables **2100** that identify, for each potential transmission path through the network, valid sets of IP addresses that can be used to transmit the packet. For example, AD table **2101** contains a plurality of IP source/destination pairs that are randomly or quasi-randomly generated. When a packet is to be transmitted from first computer **2001** to second computer **2002**, one of the link tables is randomly (or quasi-randomly) selected, and the next valid source/destination address pair from that table is used to transmit the packet through the network. If path AD is randomly selected, for example, the next source/destination IP address pair (which is pre-determined to transmit between ISP A (element **2005**) and ISP B (element **2008**)) is used to transmit the packet. If one of the transmission paths becomes degraded or inoperative, that link table can be set to a "down" condition as shown in table **2105**, thus preventing addresses from being selected from that table. Other transmission paths would be unaffected by this broken link.

3. CONTINUATION-IN-PART IMPROVEMENTS

The following describes various improvements and features that can be applied to the embodiments described above. The improvements include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities. Each is discussed separately below.

A. Load Balancer

Various embodiments described above include a system in which a transmitting node and a receiving node are coupled through a plurality of transmission paths, and wherein successive packets are distributed quasi-randomly over the plurality of paths. See, for example, FIGS. **20** and **21** and accompanying description. The improvement extends this basic concept to encompass distributing packets across different paths in such a manner that the loads on the paths are generally balanced according to transmission link quality.

In one embodiment, a system includes a transmitting node and a receiving node that are linked via a plurality of transmission paths having potentially varying transmission quality. Successive packets are transmitted over the paths based on a weight value distribution function for each path. The rate that packets will be transmitted over a given path can be different for each path. The relative "health" of each transmission path is monitored in order to identify paths that have become degraded. In one embodiment, the health of each path is monitored in the transmitter by comparing the number of packets transmitted to the number of packet acknowledgements received. Each transmission path may comprise a physically separate path (e.g., via dial-up phone line, computer network, router, bridge, or the like), or may comprise logically separate paths contained within a broadband communication medium (e.g., separate channels in an FDM, TDM, CDMA, or other type of modulated or unmodulated transmission link).

When the transmission quality of a path falls below a predetermined threshold and there are other paths that can transmit packets, the transmitter changes the weight value used for that path, making it less likely that a given packet will be transmitted over that path. The weight will preferably be set no lower than a minimum value that keeps nominal traffic on the path. The weights of the other available paths are altered to compensate for the change in the affected path. When the quality of a path degrades to where the transmitter is turned off by the synchronization function (i.e., no packets are arriving at the destination), the weight is set to zero. If all transmitters are turned off, no packets are sent.

Conventional TCP/IP protocols include a "throttling" feature that reduces the transmission rate of packets when it is determined that delays or errors are occurring in transmission. In this respect, timers are sometimes used to determine whether packets have been received. These conventional techniques for limiting transmission of packets, however, do not involve multiple transmission paths between two nodes wherein transmission across a particular path relative to the others is changed based on link quality.

According to certain embodiments, in order to damp oscillations that might otherwise occur if weight distributions are changed drastically (e.g., according to a step function), a linear or an exponential decay formula can be applied to gradually decrease the weight value over time that a degrading path will be used. Similarly, if the health of a degraded path improves, the weight value for that path is gradually increased.

Transmission link health can be evaluated by comparing the number of packets that are acknowledged within the transmission window (see embodiments discussed above) to the number of packets transmitted within that window and by the state of the transmitter (i.e., on or off). In other words, rather than accumulating general transmission statistics over time for a path, one specific implementation uses the "windowing" concepts described above to evaluate transmission path health.

The same scheme can be used to shift virtual circuit paths from an "unhealthy" path to a "healthy" one, and to select a path for a new virtual circuit.

FIG. **22**A shows a flowchart for adjusting weight values associated with a plurality of transmission links. It is assumed that software executing in one or more computer nodes executes the steps shown in FIG. **22**A. It is also assumed that the software can be stored on a computer-readable medium such as a magnetic or optical disk for execution by a computer.

Beginning in step **2201**, the transmission quality of a given transmission path is measured. As described above, this measurement can be based on a comparison between the number of packets transmitted over a particular link to the number of packet acknowledgements received over the link (e.g., per unit time, or in absolute terms). Alternatively, the quality can be evaluated by comparing the number of packets that are acknowledged within the transmission window to the number of packets that were transmitted within that window. In yet another variation, the number of missed synchronization messages can be used to indicate link quality. Many other variations are of course possible.

In step **2202**, a check is made to determine whether more than one transmitter (e.g., transmission path) is turned on. If not, the process is terminated and resumes at step **2201**.

In step **2203**, the link quality is compared to a given threshold (e.g., 50%, or any arbitrary number). If the quality falls below the threshold, then in step **2207** a check is made to determine whether the weight is above a minimum level (e.g., 1%). If not, then in step **2209** the weight is set to the minimum level and processing resumes at step **2201**. If the weight is above the minimum level, then in step **2208** the weight is gradually decreased for the path, then in step **2206** the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are increased).

If in step **2203** the quality of the path was greater than or equal to the threshold, then in step **2204** a check is made to determine whether the weight is less than a steady-state value for that path. If so, then in step **2205** the weight is increased toward the steady-state value, and in step **2206** the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are decreased). If in step **2204** the weight is not less than the steady-state value, then processing resumes at step **2201** without adjusting the weights.

The weights can be adjusted incrementally according to various functions, preferably by changing the value gradually. In one embodiment, a linearly decreasing function is used to adjust the weights; according to another embodiment, an exponential decay function is used. Gradually changing the weights helps to damp oscillators that might otherwise occur if the probabilities were abruptly.

Although not explicitly shown in FIG. **22**A the process can be performed only periodically (e.g., according to a time schedule), or it can be continuously run, such as in a background mode of operation. In one embodiment, the combined weights of all potential paths should add up to unity (e.g., when the weighting for one path is decreased, the corresponding weights that the other paths will be selected will increase).

Adjustments to weight values for other paths can be prorated. For example, a decrease of 10% in weight value for one path could result in an evenly distributed increase in the weights for the remaining paths. Alternatively, weightings could be adjusted according to a weighted formula as desired (e.g., favoring healthy paths over less healthy paths). In yet another variation, the difference in weight value can be amortized over the remaining links in a manner that is proportional to their traffic weighting.

FIG. **22**B shows steps that can be executed to shut down transmission links where a transmitter turns off. In step **2210**, a transmitter shut-down event occurs. In step **2211**, a test is made to determine whether at least one transmitter is still turned on. If not, then in step **2215** all packets are dropped until a transmitter turns on. If in step **2211** at least one transmitter is turned on, then in step **2212** the weight for the path is set to zero, and the weights for the remaining paths are adjusted accordingly.

FIG. **23** shows a computer node **2301** employing various principles of the above-described embodiments. It is assumed that two computer nodes of the type shown in FIG. **23** communicate over a plurality of separate physical transmission paths. As shown in FIG. **23**, four transmission paths X1 through X4 are defined for communicating between the two nodes. Each node includes a packet transmitter **2302** that operates in accordance with a transmit table **2308** as described above. (The packet transmitter could also operate without using the IP-hopping features described above, but the following description assumes that some form of hopping is employed in conjunction with the path selection mechanism.). The computer node also includes a packet receiver **2303** that operates in accordance with a receive table **2309**, including a moving window W that moves as valid packets are received. Invalid packets having source and destination addresses that do not fall within window W are rejected.

As each packet is readied for transmission, source and destination IP addresses (or other discriminator values) are selected from transmit table **2308** according to any of the various algorithms described above, and packets containing these source/destination address pairs, which correspond to the node to which the four transmission paths are linked, are generated to a transmission path switch **2307**. Switch **2307**, which can comprise a software function, selects from one of the available transmission paths according to a weight distribution table **2306**. For example, if the weight for path X1 is 0.2, then every fifth packet will be transmitted on path X**1**. A similar regime holds true for the other paths as shown. Initially, each link's weight value can be set such that it is proportional to its bandwidth, which will be referred to as its "steady-state" value.

Packet receiver **2303** generates an output to a link quality measurement function **2304** that operates as described above to determine the quality of each transmission path. (The input to packet receiver **2303** for receiving incoming packets is omitted for clarity). Link quality measurement function **2304** compares the link quality to a threshold for each transmission link and, if necessary, generates an output to weight adjustment function **2305**. If a weight adjustment is required, then the weights in table **2306** are adjusted accordingly, preferably according to a gradual (e.g., linearly or exponentially declining) function. In one embodiment, the weight values for all available paths are initially set to the same value, and only when paths degrade in quality are the weights changed to reflect differences.

Link quality measurement function **2304** can be made to operate as part of a synchronizer function as described above. That is, if resynchronization occurs and the receiver detects that synchronization has been lost (e.g., resulting in the synchronization window W being advanced out of sequence), that fact can be used to drive link quality measurement function **2304**. According to one embodiment, load balancing is performed using information garnered during the normal synchronization, augmented slightly to communicate link health from the receiver to the transmitter. The receiver maintains a count, MESS_R(W), of the messages received in synchronization window W. When it receives a synchronization request (SYNC_REQ) corresponding to the end of window W, the receiver includes counter MESS_R in the resulting synchronization acknowledgement (SYNC_ACK) sent back to the transmitter. This allows the transmitter to compare messages sent to messages received in order to asses the health of the link.

If synchronization is completely lost, weight adjustment function **2305** decreases the weight value on the affected path to zero. When synchronization is regained, the weight value for the affected path is gradually increased to its original value. Alternatively, link quality can be measured by evaluating the length of time required for the receiver to acknowledge a synchronization request. In one embodiment, separate transmit and receive tables are used for each transmission path.

When the transmitter receives a SYNC_ACK, the MESS_R is compared with the number of messages transmitted in a window (MESS_T). When the transmitter receives a SYNC_ACK, the traffic probabilities will be examined and adjusted if necessary. MESS_R is compared with the number of messages transmitted in a window (MESS_T). There are two possibilities:

1. If MESS_R is less than a threshold value, THRESH, then the link will be deemed to be unhealthy. If the transmitter was turned off, the transmitter is turned on and the weight P for that link will be set to a minimum value MIN. This will keep a trickle of traffic on the link for monitoring purposes until it recovers. If the transmitter was turned on, the weight P for that link will be set to:

$$P'=\alpha\times MIN+(1-\alpha)\times P \qquad (1)$$

Equation 1 will exponentially damp the traffic weight value to MIN during sustained periods of degraded service.

2. If MESS_R for a link is greater than or equal to THRESH, the link will be deemed healthy. If the weight P for that link is greater than or equal to the steady state value S for that link, then P is left unaltered. If the weight P for that link is less than THRESH then P will be set to:

$$P'=\beta\times S+(1-\beta)\times P \qquad (2)$$

where β is a parameter such that 0<=β<=1 that determines the damping rate of P.

Equation 2 will increase the traffic weight to S during sustained periods of acceptable service in a damped exponential fashion.

A detailed example will now be provided with reference to FIG. **24**. As shown in FIG. **24**, a first computer **2401** communicates with a second computer **2402** through two routers **2403** and **2404**. Each router is coupled to the other router through three transmission links. As described above, these may be physically diverse links or logical links (including virtual private networks).

Suppose that a first link L**1** can sustain a transmission bandwidth of 100 Mb/s and has a window size of 32; link L**2** can sustain 75 Mb/s and has a window size of 24; and link L**3** can sustain 25 Mb/s and has a window size of 8. The combined links can thus sustain 200Mb/s. The steady state traffic weights are 0.5 for link L**1**; 0.375 for link L**2**, and 0.125 for link L**3**. MIN=1Mb/s, THRESH=0.8 MESS_T for each link, α=0.75 and β=0.5. These traffic weights will remain stable until a link stops for synchronization or reports a number of packets received less than its THRESH. Consider the following sequence of events:

1. Link L**1** receives a SYNC_ACK containing a MESS_R of 24, indicating that only 75% of the MESS_T (32) messages transmitted in the last window were successfully received. Link **1** would be below THRESH (0.8). Consequently, link L**1**'s traffic weight value would be reduced to 0.12825, while link L**2**'s traffic weight value would be increased to 0.65812 and link L**3**'s traffic weight value would be increased to 0.217938.
2. Link L**2** and L**3** remained healthy and link L**1** stopped to synchronize. Then link L**1**'s traffic weight value would be set to 0, link L**2**'s traffic weight value would be set to 0.75, and link L**33**'s traffic weight value would be set to 0.25.
3. Link L**1** finally received a SYNC_ACK containing a MESS_R of 0 indicating that none of the MESS_T (32) messages transmitted in the last window were successfully received. Link L**1** would be below THRESH. Link L**1**'s traffic weight value would be increased to 0.005, link L**2**'s traffic weight value would be decreased to 0.74625, and link L**3**'s traffic weight value would be decreased to 0.24875.
4. Link L**1** received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L**1** would be above THRESH. Link L**1**'s traffic weight value would be increased to 0.2525, while link L**2**'s traffic weight value would be decreased to 0.560625 and link L**3**'s traffic weight value would be decreased to 0.186875.
5. Link L**1** received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L**1** would be above THRESH. Link L**1**'s traffic weight value would be increased to 0.37625; link L**2**'s traffic weight value would be decreased to 0.4678125, and link L**3**'s traffic weight value would be decreased to 0.1559375.
6. Link L**1** remains healthy and the traffic probabilities approach their steady state traffic probabilities.

B. Use of a DNS Proxy to Transparently Create Virtual Private Networks

A second improvement concerns the automatic creation of a virtual private network (VPN) in response to a domain-name server look-up function.

Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host. For example, when a computer user types in the web name "Yahoo.com," the user's web browser transmits a request to a DNS, which converts the name into a four-part IP address that is returned to the user's browser and then used by the browser to contact the destination web site.

This conventional scheme is shown in FIG. **25**. A user's computer **2501** includes a client application **2504** (for example, a web browser) and an IP protocol stack **2505**. When the user enters the name of a destination host, a request DNS REQ is made (through IP protocol stack **2505**) to a DNS **2502** to look up the IP address associated with the name. The DNS returns the IP address DNS RESP to client application **2504**, which is then able to use the IP address to communicate with the host **2503** through separate transactions such as PAGE REQ and PAGE RESP.

In the conventional architecture shown in FIG. **25**, nefarious listeners on the Internet could intercept the DNS REQ and DNS RESP packets and thus learn what IP addresses the user was contacting. For example, if a user wanted to set up a secure communication path with a web site having the name "Target.com," when the user's browser contacted a DNS to find the IP address for that web site, the true IP address of that web site would be revealed over the Internet as part of the DNS inquiry. This would hamper anonymous communications on the Internet.

One conventional scheme that provides secure virtual private networks over the Internet provides the DNS server with the public keys of the machines that the DNS server has the addresses for. This allows hosts to retrieve automatically the public keys of a host that the host is to communicate with so that the host can set up a VPN without having the user enter the public key of the destination host. One implementation of this standard is presently being developed as part of the FreeS/WAN project(RFC **2535**).

The conventional scheme suffers from certain drawbacks. For example, any user can perform a DNS request. Moreover, DNS requests resolve to the same value for all users.

According to certain aspects of the invention, a specialized DNS server traps DNS requests and, if the request is from a special type of user (e.g., one for which secure communication services are defined), the server does not return the true IP address of the target node, but instead automatically sets up a virtual private network between the target node and the user. The VPN is preferably implemented using the IP address "hopping" features of the basic invention described above, such that the true identity of the two nodes cannot be determined even if packets during the communication are intercepted. For DNS requests that are determined to not require secure services (e.g., an unregistered user), the DNS server transparently "passes through" the request to provide a normal look-up function and return the IP address of the target web server, provided that the requesting host has permissions to resolve unsecured sites. Different users who make an identical DNS request could be provided with different results.

FIG. **26** shows a system employing various principles summarized above. A user's computer **2601** includes a conventional client (e.g., a web browser) **2605** and an IP protocol stack **2606** that preferably operates in accordance with an IP hopping function **2607** as outlined above. A modified DNS server **2602** includes a conventional DNS server function **2609** and a DNS proxy **2610**. A gatekeeper server **2603** is interposed between the modified DNS server and a secure target site **2704**. An "unsecure" target site **2611** is also accessible via conventional IP protocols.

According to one embodiment, DNS proxy **2610** intercepts all DNS lookup functions from client **2605** and determines whether access to a secure site has been requested. If access to a secure site has been requested (as determined, for example, by a domain name extension, or by reference to an internal table of such sites), DNS proxy **2610** determines whether the user has sufficient security privileges to access the site. If so, DNS proxy **2610** transmits a message to gatekeeper **2603** requesting that a virtual private network be created between user computer **2601** and secure target site **2604**. In one embodiment, gatekeeper **2603** creates "hopblocks" to be used by computer **2601** and secure target site **2604** for secure communication. Then, gatekeeper **2603** communicates these to user computer **2601**. Thereafter, DNS proxy **2610** returns to user computer **2601** the resolved address passed to it by the gatekeeper (this address could be different from the actual target computer) **2604**, preferably using a secure administrative VPN. The address that is returned need not be the actual address of the destination computer.

Had the user requested lookup of a non-secure web site such as site **2611**, DNS proxy would merely pass through to conventional DNS server **2609** the look-up request, which would be handled in a conventional manner, returning the IP address of non-secure web site **2611**. If the user had requested lookup of a secure web site but lacked credentials to create such a connection, DNS proxy **2610** would return a "host unknown" error to the user. In this manner, different users requesting access to the same DNS name could be provided with different look-up results.

Gatekeeper **2603** can be implemented on a separate computer (as shown in FIG. **26**) or as a function within modified DNS server **2602**. In general, it is anticipated that gatekeeper **2703** facilitates the allocation and exchange of information needed to communicate securely, such as using "hopped" IP addresses. Secure hosts such as site **2604** are assumed to be equipped with a secure communication function such as an IP hopping function **2608**.

It will be appreciated that the functions of DNS proxy **2610** and DNS server **2609** can be combined into a single server for convenience. Moreover, although element **2602** is shown as combining the functions of two servers, the two servers can be made to operate independently.

FIG. **27** shows steps that can be executed by DNS proxy server **2610** to handle requests for DNS look-up for secure hosts. In step **2701**, a DNS look-up request is received for a target host. In step **2702**, a check is made to determine whether access to a secure host was requested. If not, then in step **2703** the DNS request is passed to conventional DNS server **2609**, which looks up the IP address of the target site and returns it to the user's application for further processing.

In step **2702**, if access to a secure host was requested, then in step **2704** a further check is made to determine whether the user is authorized to connect to the secure host. Such a check can be made with reference to an internally stored list of authorized IP addresses, or can be made by communicating with gatekeeper **2603** (e.g., over an "administrative" VPN that is secure). It will be appreciated that different levels of security can also be provided for different categories of hosts. For example, some sites may be designated as having a certain security level, and the security level of the user requesting access must match that security level. The user's security level can also be determined by transmitting a request message back to the user's computer requiring that it prove that it has sufficient privileges.

If the user is not authorized to access the secure site, then a "host unknown" message is returned (step **2705**). If the user has sufficient security privileges, then in step **2706** a secure VPN is established between the user's computer and the secure target site. As described above, this is preferably done by allocating a hopping regime that will be carried out between the user's computer and the secure target site, and is preferably performed transparently to the user (i.e., the user need not be involved in creating the secure link). As described in various embodiments of this application, any of various fields can be "hopped" (e.g., IP source/destination addresses; a field in the header; etc.) in order to communicate securely.

Some or all of the security functions can be embedded in gatekeeper **2603**, such that it handles all requests to connect to secure sites. In this embodiment, DNS proxy **2610** communicates with gatekeeper **2603** to determine (preferably over a secure administrative VPN) whether the user has access to a particular web site. Various scenarios for implementing these features are described by way of example below:

Scenario #1: Client has permission to access target computer, and gatekeeper has a rule to make a VPN for the client. In this scenario, the client's DNS request would be received by the DNS proxy server **2610**, which would forward the request to gatekeeper **2603**. The gatekeeper would establish a VPN between the client and the requested target. The gatekeeper would provide the address of the destination to the DNS proxy, which would then return the resolved name as a result. The resolved address can be transmitted back to the client in a secure administrative VPN.

Scenario #2: Client does not have permission to access target computer. In this scenario, the client's DNS request would be received by the DNS proxy server **2610**, which would forward the request to gatekeeper **2603**. The gatekeeper would reject the request, informing DNS proxy server **2610** that it was unable to find the target computer. The DNS proxy **2610** would then return a "host unknown" error message to the client.

Scenario #3: Client has permission to connect using a normal non-VPN link, and the gatekeeper does not have a rule to set up a VPN for the client to the target site. In this scenario, the client's DNS request is received by DNS proxy server **2610**, which would check its rules and determine that no VPN is needed. Gatekeeper **2603** would then inform the DNS proxy server to forward the request to conventional DNS server **2609**, which would resolve the request and return the result to the DNS proxy server and then back to the client.

Scenario #4: Client does not have permission to establish a normal/non-VPN link, and the gatekeeper does not have a rule to make a VPN for the client to the target site. In this scenario, the DNS proxy server would receive the client's DNS request and forward it to gatekeeper **2603**. Gatekeeper **2603** would determine that no special VPN was needed, but that the client is not authorized to communicate with non-VPN members. The gatekeeper would reject the request, causing DNS proxy server **2610** to return an error message to the client.

C. Large Link to Small Link Bandwidth Management

One feature of the basic architecture is the ability to prevent so-called "denial of service" attacks that can occur if a computer hacker floods a known Internet node with packets, thus preventing the node from communicating with other nodes. Because IP addresses or other fields are "hopped" and packets arriving with invalid addresses are quickly discarded, Internet nodes are protected against flooding targeted at a single IP address.

In a system in which a computer is coupled through a link having a limited bandwidth (e.g., an edge router) to a node that can support a much higher-bandwidth link (e.g., an Internet Service Provider), a potential weakness could be exploited by a determined hacker. Referring to FIG. **28**, suppose that a first host computer **2801** is communicating with a second host computer **2804** using the IP address hopping principles described above. The first host computer is coupled through an edge router **2802** to an Internet Service Provider (ISP) **2803** through a low bandwidth link (LOW BW), and is in turn coupled to second host computer **2804** through parts of the Internet through a high bandwidth link (HIGH BW). In this architecture, the ISP is able to support a high bandwidth to the internet, but a much lower bandwidth to the edge router **2802**.

Suppose that a computer hacker is able to transmit a large quantity of dummy packets addressed to first host computer **2801** across high bandwidth link HIGH BW. Normally, host computer **2801** would be able to quickly reject the packets since they would not fall within the acceptance window permitted by the IP address hopping scheme. However, because the packets must travel across low bandwidth link LOW BW, the packets overwhelm the lower bandwidth link before they are received by host computer **2801**. Consequently, the link to host computer **2801** is effectively flooded before the packets can be discarded.

According to one inventive improvement, a "link guard" function **2805** is inserted into the high-bandwidth node (e.g., ISP **2803**) that quickly discards packets destined for a low-bandwidth target node if they are not valid packets. Each packet destined for a low-bandwidth node is cryptographically authenticated to determine whether it belongs to a VPN. If it is not a valid VPN packet, the packet is discarded at the high-bandwidth node. If the packet is authenticated as belonging to a VPN, the packet is passed with high preference. If the packet is a valid non-VPN packet, it is passed with a lower quality of service (e.g., lower priority).

In one embodiment, the ISP distinguishes between VPN and non-VPN packets using the protocol of the packet. In the case of IPSEC [rfc **2401**], the packets have IP protocols **420** and **421**. In the case of the TARP VPN, the packets will have an IP protocol that is not yet defined. The ISP's link guard, **2805**, maintains a table of valid VPNs which it uses to validate whether VPN packets are cryptographically valid.

According to one embodiment, packets that do not fall within any hop windows used by nodes on the low-bandwidth link are rejected, or are sent with a lower quality of service. One approach for doing this is to provide a copy of the IP hopping tables used by the low-bandwidth nodes to the high-bandwidth node, such that both the high-bandwidth and low-bandwidth nodes track hopped packets (e.g., the high-bandwidth node moves its hopping window as valid packets are received). In such a scenario, the high-bandwidth node discards packets that do not fall within the hopping window before they are transmitted over the low-bandwidth link. Thus, for example, ISP **2903** maintains a copy **2910** of the receive table used by host computer **2901**. Incoming packets that do not fall within this receive table are discarded. According to a different embodiment, link guard **2805** validates each VPN packet using a keyed hashed message authentication code (HMAC) [rfc **2104**]. According to another embodiment, separate VPNs (using, for example, hopblocks) can be established for communicating between the low-bandwidth node and the high-bandwidth node (i.e., packets arriving at the high-bandwidth node are converted into different packets before being transmitted to the low-bandwidth node).

As shown in FIG. **29**, for example, suppose that a first host computer **2900** is communicating with a second host computer **2902** over the Internet, and the path includes a high bandwidth link HIGH BW to an ISP **2901** and a low bandwidth link LOW BW through an edge router **2904**. In accordance with the basic architecture described above, first host computer **2900** and second host computer **2902** would exchange hopblocks (or a hopblock algorithm) and would be able to create matching transmit and receive tables **2905**, **2906**, **2912** and **2913**. Then in accordance with the basic architecture, the two computers would transmit packets having seemingly random IP source and destination addresses, and each would move a corresponding hopping window in its receive table as valid packets were received.

Suppose that a nefarious computer hacker **2903** was able to deduce that packets having a certain range of IP addresses (e.g., addresses 100 to 200 for the sake of simplicity) are being transmitted to ISP **2901**, and that these packets are being forwarded over a low-bandwidth link. Hacker computer **2903** could thus "flood" packets having addresses falling into the range 100 to 200, expecting that they would be forwarded along low bandwidth link LOW BW, thus causing the low bandwidth link to become overwhelmed. The fast packet reject mechanism in first host computer **3000** would be of little use in rejecting these packets, since the low bandwidth link was effectively jammed before the packets could be rejected. In accordance with one aspect of the improvement, however, VPN link guard **2911** would prevent the attack from impacting the performance of VPN traffic because the packets would either be rejected as invalid VPN packets or given a lower quality of service than VPN traffic over the lower bandwidth link. A denial-of-service flood attack could, however, still disrupt non-VPN traffic.

According to one embodiment of the improvement, ISP **2901** maintains a separate VPN with first host computer **2900**, and thus translates packets arriving at the ISP into packets having a different IP header before they are transmitted to host computer **2900**. The cryptographic keys used to authenticate VPN packets at the link guard **2911** and the cryptographic keys used to encrypt and decrypt the VPN packets at host **2902** and host **2901** can be different, so that link guard **2911** does not have access to the private host data; it only has the capability to authenticate those packets.

According to yet a third embodiment, the low-bandwidth node can transmit a special message to the high-bandwidth node instructing it to shut down all transmissions on a particular IP address, such that only hopped packets will pass through to the low-bandwidth node. This embodiment would prevent a hacker from flooding packets using a single IP address. According to yet a fourth embodiment, the high-bandwidth node can be configured to discard packets transmitted to the low-bandwidth node if the transmission rate exceeds a certain predetermined threshold for any given IP address; this would allow hopped packets to go through. In this respect, link guard **2911** can be used to detect that the rate of packets on a given IP address are exceeding a threshold rate; further packets addressed to that same IP address would be dropped or transmitted at a lower priority (e.g., delayed).

D. Traffic Limiter

In a system in which multiple nodes are communicating using "hopping" technology, a treasonous insider could internally flood the system with packets. In order to prevent this possibility, one inventive improvement involves setting up "contracts" between nodes in the system, such that a receiver can impose a bandwidth limitation on each packet sender. One technique for doing this is to delay acceptance of a checkpoint synchronization request from a sender until a certain time period (e.g., one minute) has elapsed. Each receiver can effectively control the rate at which its hopping window moves by delaying "SYNC ACK" responses to "SYNC_REQ" messages.

A simple modification to the checkpoint synchronizer will serve to protect a receiver from accidental or deliberate overload from an internally treasonous client. This modification is based on the observation that a receiver will not update its

tables until a SYNC_REQ is received on hopped address CKPT_N. It is a simple matter of deferring the generation of a new CKPT_N until an appropriate interval after previous checkpoints.

Suppose a receiver wished to restrict reception from a transmitter to 100 packets a second, and that checkpoint synchronization messages were triggered every 50 packets. A compliant transmitter would not issue new SYNC_REQ messages more often than every 0.5 seconds. The receiver could delay a non-compliant transmitter from synchronizing by delaying the issuance of CKPT_N for 0.5 second after the last SYNC_REQ was accepted.

In general, if M receivers need to restrict N transmitters issuing new SYNC_REQ messages after every W messages to sending R messages a second in aggregate, each receiver could defer issuing a new CKPT_N until M×N×W/R seconds have elapsed since the last SYNC_REQ has been received and accepted. If the transmitter exceeds this rate between a pair of checkpoints, it will issue the new checkpoint before the receiver is ready to receive it, and the SYNC_REQ will be discarded by the receiver. After this, the transmitter will reissue the SYNC_REQ every T**1** seconds until it receives a SYNC_ACK. The receiver will eventually update CKPT_N and the SYNC_REQ will be acknowledged. If the transmission rate greatly exceeds the allowed rate, the transmitter will stop until it is compliant. If the transmitter exceeds the allowed rate by a little, it will eventually stop after several rounds of delayed synchronization until it is in compliance. Hacking the transmitter's code to not shut off only permits the transmitter to lose the acceptance window. In this case it can recover the window and proceed only after it is compliant again.

Two practical issues should be considered when implementing the above scheme:

1. The receiver rate should be slightly higher than the permitted rate in order to allow for statistical fluctuations in traffic arrival times and non-uniform load balancing.

2. Since a transmitter will rightfully continue to transmit for a period after a SYNC_REQ is transmitted, the algorithm above can artificially reduce the transmitter's bandwidth. If events prevent a compliant transmitter from synchronizing for a period (e.g. the network dropping a SYNC_REQ or a SYNC_ACK) a SYNC_REQ will be accepted later than expected. After this, the transmitter will transmit fewer than expected messages before encountering the next checkpoint. The new checkpoint will not have been activated and the transmitter will have to retransmit the SYNC_REQ. This will appear to the receiver as if the transmitter is not compliant. Therefore, the next checkpoint will be accepted late from the transmitter's perspective. This has the effect of reducing the transmitter's allowed packet rate until the transmitter transmits at a packet rate below the agreed upon rate for a period of time.

To guard against this, the receiver should keep track of the times that the last C SYNC_REQs were received and accepted and use the minimum of M×N×W/R seconds after the last SYNC_REQ has been received and accepted, 2×M×N×W/R seconds after next to the last SYNC_REQ has been received and accepted, C×M×N×W/R seconds after $(C-1)^{th}$ to the last SYNC_REQ has been received, as the time to activate CKPT_N. This prevents the receiver from inappropriately limiting the transmitter's packet rate if at least one out of the last C SYNC_REQs was processed on the first attempt.

FIG. **30** shows a system employing the above-described principles. In FIG. **30**, two computers **3000** and **3001** are assumed to be communicating over a network N in accordance with the "hopping" principles described above (e.g., hopped IP addresses, discriminator values, etc.). For the sake of simplicity, computer **3000** will be referred to as the receiving computer and computer **3001** will be referred to as the transmitting computer, although full duplex operation is of course contemplated. Moreover, although only a single transmitter is shown, multiple transmitters can transmit to receiver **3000**.

As described above, receiving computer **3000** maintains a receive table **3002** including a window W that defines valid IP address pairs that will be accepted when appearing in incoming data packets. Transmitting computer **3001** maintains a transmit table **3003** from which the next IP address pairs will be selected when transmitting a packet to receiving computer **3000**. (For the sake of illustration, window W is also illustrated with reference to transmit table **3003**). As transmitting computer moves through its table, it will eventually generate a SYNC_REQ message as illustrated in function **3010**. This is a request to receiver **3000** to synchronize the receive table **3002**, from which transmitter **3001** expects a response in the form of a CKPT_N (included as part of a SYNC_ACK message). If transmitting computer **3001** transmits more messages than its allotment, it will prematurely generate the SYNC_REQ message. (If it has been altered to remove the SYNC_REQ message generation altogether, it will fall out of synchronization since receiver **3000** will quickly reject packets that fall outside of window W, and the extra packets generated by transmitter **3001** will be discarded).

In accordance with the improvements described above, receiving computer **3000** performs certain steps when a SYNC_REQ message is received, as illustrated in FIG. **30**. In step **3004**, receiving computer **3000** receives the SYNC_REQ message. In step **3005**, a check is made to determine whether the request is a duplicate. If so, it is discarded in step **3006**. In step **3007**, a check is made to determine whether the SYNC_REQ received from transmitter **3001** was received at a rate that exceeds the allowable rate R (i.e., the period between the time of the last SYNC_REQ message). The value R can be a constant, or it can be made to fluctuate as desired. If the rate exceeds R, then in step **3008** the next activation of the next CKPT_N hopping table entry is delayed by W/R seconds after the last SYNC_REQ has been accepted.

Otherwise, if the rate has not been exceeded, then in step **3109** the next CKPT_N value is calculated and inserted into the receiver's hopping table prior to the next SYNC_REQ from the transmitter **3101**. Transmitter **3101** then processes the SYNC_REQ in the normal manner.

E. Signaling Synchronizer

In a system in which a large number of users communicate with a central node using secure hopping technology, a large amount of memory must be set aside for hopping tables and their supporting data structures. For example, if one million subscribers to a web site occasionally communicate with the web site, the site must maintain one million hopping tables, thus using up valuable computer resources, even though only a small percentage of the users may actually be using the system at any one time. A desirable solution would be a system that permits a certain maximum number of simultaneous links to be maintained, but which would "recognize" millions of registered users at any one time. In other words, out of a population of a million registered users, a few thousand at a time could simultaneously communicate with a central server, without requiring that the server maintain one million hopping tables of appreciable size.

One solution is to partition the central node into two nodes: a signaling server that performs session initiation for user log-on and log-off (and requires only minimally sized tables), and a transport server that contains larger hopping tables for the users. The signaling server listens for the millions of known users and performs a fast-packet reject of other (bogus) packets. When a packet is received from a known user, the signaling server activates a virtual private link (VPL) between the user and the transport server, where hopping tables are allocated and maintained. When the user logs onto the signaling server, the user's computer is provided with hop tables for communicating with the transport server, thus activating the VPL. The VPLs can be torn down when they become inactive for a time period, or they can be torn down upon user log-out. Communication with the signaling server to allow user log-on and log-off can be accomplished using a specialized version of the checkpoint scheme described above.

FIG. **31** shows a system employing certain of the above-described principles. In FIG. **31**, a signaling server **3101** and a transport server **3102** communicate over a link. Signaling server **3101** contains a large number of small tables **3106** and **3107** that contain enough information to authenticate a communication request with one or more clients **3103** and **3104**. As described in more detail below, these small tables may advantageously be constructed as a special case of the synchronizing checkpoint tables described previously. Transport server **3102**, which is preferably a separate computer in communication with signaling server **3101**, contains a smaller number of larger hopping tables **3108**, **3109**, and **3110** that can be allocated to create a VPN with one of the client computers.

According to one embodiment, a client that has previously registered with the system (e.g., via a system administration function, a user registration procedure, or some other method) transmits a request for information from a computer (e.g., a web site). In one variation, the request is made using a "hopped" packet, such that signaling server **3101** will quickly reject invalid packets from unauthorized computers such as hacker computer **3105**. An "administrative" VPN can be established between all of the clients and the signaling server in order to ensure that a hacker cannot flood signaling server **3101** with bogus packets. Details of this scheme are provided below.

Signaling server **3101** receives the request **3111** and uses it to determine that client **3103** is a validly registered user. Next, signaling server **3101** issues a request to transport server **3102** to allocate a hopping table (or hopping algorithm or other regime) for the purpose of creating a VPN with client **3103**. The allocated hopping parameters are returned to signaling server **3101** (path **3113**), which then supplies the hopping parameters to client **3103** via path **3114**, preferably in encrypted form.

Thereafter, client **3103** communicates with transport server **3102** using the normal hopping techniques described above. It will be appreciated that although signaling server **3101** and transport server **3102** are illustrated as being two separate computers, they could of course be combined into a single computer and their functions performed on the single computer. Alternatively, it is possible to partition the functions shown in FIG. **31** differently from as shown without departing from the inventive principles.

One advantage of the above-described architecture is that signaling server **3101** need only maintain a small amount of information on a large number of potential users, yet it retains the capability of quickly rejecting packets from unauthorized users such as hacker computer **3105**. Larger data tables needed to perform the hopping and synchronization functions are instead maintained in a transport server **3102**, and a smaller number of these tables are needed since they are only allocated for "active" links. After a VPN has become inactive for a certain time period (e.g., one hour), the VPN can be automatically torn down by transport server **3102** or signaling server **3101**.

A more detailed description will now be provided regarding how a special case of the checkpoint synchronization feature can be used to implement the signaling scheme described above.

The signaling synchronizer may be required to support many (millions) of standing, low bandwidth connections. It therefore should minimize per-VPL memory usage while providing the security offered by hopping technology. In order to reduce memory usage in the signaling server, the data hopping tables can be completely eliminated and data can be carried as part of the SYNC_REQ message. The table used by the server side (receiver) and client side (transmitter) is shown schematically as element **3106** in FIG. **31**.

The meaning and behaviors of CKPT_N, CKPT_O and CKPT_R remain the same from the previous description, except that CKPT_N can receive a combined data and SYNC_REQ message or a SYNC_REQ message without the data.

The protocol is a straightforward extension of the earlier synchronizer. Assume that a client transmitter is on and the tables are synchronized. The initial tables can be generated "out of band." For example, a client can log into a web server to establish an account over the Internet. The client will receive keys etc encrypted over the Internet. Meanwhile, the server will set up the signaling VPN on the signaling server.

Assuming that a client application wishes to send a packet to the server on the client's standing signaling VPL:

1. The client sends the message marked as a data message on the inner header using the transmitter's CKPT_N address. It turns the transmitter off and starts a timer T**1** noting CKPT_O. Messages can be one of three types: DATA, SYNC_REQ and SYNC_ACK. In the normal algorithm, some potential problems can be prevented by identifying each message type as part of the encrypted inner header field. In this algorithm, it is important to distinguish a data packet and a SYNC_REQ in the signaling synchronizer since the data and the SYNC_REQ come in on the same address.
2. When the server receives a data message on its CKPT_N, it verifies the message and passes it up the stack. The message can be verified by checking message type and and other information (i.e user credentials) contained in the inner header It replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side

CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

3. When the client side receiver receives a SYNC_ACK on its CKPT_R with a payload matching its transmitter side CKPT_O and the transmitter is off, the transmitter is turned on and the receiver side CKPT_R is updated. If the SYNC_ACK's payload does not match the transmitter side CKPT_O or the transmitter is on, the SYNC_ACK is simply discarded.
4. T**1** expires: If the transmitter is off and the client's transmitter side CKPT_O matches the CKPT_O associated with the timer, it starts timer T1 noting CKPT_O again, and a SYNC_REQ is sent using the transmitter's CKPT_O address. Otherwise, no action is taken.
5. When the server receives a SYNC_REQ on its CKPT_N, it replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.
6. When the server receives a SYNC_REQ on its CKPT_O, it updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

FIG. **32** shows message flows to highlight the protocol. Reading from top to bottom, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is successfully received and a passed up the stack. It also synchronizes the receiver i.e, the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates a new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned on and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

Next, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is lost. The client side timer expires and as a result a SYNC_REQ is transmitted on the client side transmitter's CKPT_O (this will keep happening until the SYNC_ACK has been received at the client). The SYNC_REQ is successfully received at the server. It synchronizes the receiver i.e, the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates an new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned off and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

There are numerous other scenarios that follow this flow. For example, the SYNC_ACK could be lost. The transmitter would continue to re-send the SYNC_REQ until the receiver synchronizes and responds.

The above-described procedures allow a client to be authenticated at signaling server **3201** while maintaining the ability of signaling server **3201** to quickly reject invalid packets, such as might be generated by hacker computer **3205**. In various embodiments, the signaling synchronizer is really a derivative of the synchronizer. It provides the same protection as the hopping protocol, and it does so for a large number of low bandwidth connections.

We claim:

1. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

2. The data processing device of claim **1**, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

3. The data processing device of claim **2**, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

4. The data processing device of claim **3**, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

5. The data processing device of claim **1**, wherein automatically initiating the encrypted channel between the client and the secure sewer comprises establishing an IF address hopping scheme between the client and the secure server.

6. The data processing device of claim **1**, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

7. A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

8. The computer readable medium of claim **7**, wherein step (iv) comprises the steps of

(a) determining whether the client is authorized to access the secure server, and

(b) when the client is authorized to access the secure server, sending a request to the secure sewer to establish an encrypted channel between the secure sewer and the client.

9. The computer readable medium of claim **8**, wherein step (iv) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

10. The computer readable medium of claim **9**, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

11. The computer readable medium of claim **7**, wherein automatically initiating the encrypted channel between the client and the secure sewer comprises establishing an IP address hopping scheme between the client and the secure server.

12. The computer readable medium of claim **7**, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

13. A computer readable medium storing a domain name server (DNS) module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) determining whether a DNS request sent by a client corresponds to a secure server;

(ii) when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iii) when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel between the client and the secure server.

14. The computer readable medium of claim **13**, wherein step (iii) comprises the steps of

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish a secure channel between the secure server and the client.

15. The computer readable medium of claim **14**, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

16. The computer readable medium of claim **15**, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

* * * * *

CERTIFICATE OF SERVICE

I, Naveen Modi, hereby certify that on February 5, 2024, the foregoing brief was filed using the Court's CM/ECF system and a copy served on the parties' counsel of record via ECF.

Date: February 5, 2024 BY: /s/Naveen Modi

Naveen Modi
PAUL HASTINGS LLP
2050 M Street, N.W.
Washington, D.C. 20036
Tel.: (202) 551-1700
Fax: (202) 551-1705

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS AND TYPE STYLE REQUIREMENTS

1. This brief complies with the type-volume limitation of Federal Circuit Rule 32 and Federal Circuit Rule 28.1(b)(1)(A).

The brief contains 12,823 words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(f) and Federal Circuit Rule 32(b).

2. This brief complies with the typeface requirements of Federal Rule of Appellate Procedure 32(a)(5) and the type style requirements of Federal Rule of Appellate Procedure 32(a)(6).

The brief has been prepared in a proportionally spaced typeface using MS Word 2013 in a 14 point Times New Roman font.

Date: February 5, 2024 BY: /s/Naveen Modi

Naveen Modi
PAUL HASTINGS LLP
2050 M Street, N.W.
Washington, D.C. 20036
Tel.: (202) 551-1700
Fax: (202) 551-1705